Learning to Measure: Distance Metric Learning with Structured Sparsity

By

Joseph St.Amand

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Committee members

Dr. Arvin Agah, Chairperson

Dr. James Miller

Dr. Prasad Kulkarni

Dr. Guanghui Wang

Dr. Bozenna Pasik-Duncan

Date defended: April 13, 2018

The Thesis Committee for Joseph St.Amand certifies
that this is the approved version of the following thesis :

Learning to Measure: Distance Metric Learning with Structured Sparsity

Dr. Arvin Agah, Chairperson

Date approved: April 13, 2018

Acknowledgements

This dissertation represents a major milestone in a process of lifelong learning and is something that in my early years I found difficult to believe I would ever achieve. For myself, the journey has been long and arduous, with many difficulties along the way. Reflecting back on this experience there are dozens of people who have contributed to my personal development and motivated my pursuit of this academic summit.

During my time at KU I had a knack for selecting advisers who were soon to be moving on to bigger and better things. First and foremost of which I would like to thank Dr. Jun (Luke) Huan, with whom I spent the majority of my time at KU with. Dr. Huan's insights into the research process, especially how to read, write, and review papers, is something that will stay with me for a lifetime. I'd also like to thank my first year adviser Dr. Brian Potetz, although our time together was short, I received a disproportionately large amount of advice and support from him whenever I thought I was in over my head. I finished my studies with Dr. Arvin Agah, whose responsiveness and positive attitude has propelled me through the final stages. Finally, I'd like to thank each of my committee members, Dr. Arvin Agah, Dr. Prasad Kulkarni, Dr. James Miller, Dr. Richard Wang, and Dr. Bozenna Pasik-Duncan, each of whom has always been available anytime I've requested their input.

A majority of my semesters were spent working as a graduate teaching assistant, I'd like to express my thanks to the Electrical Engineering and Computer Science (EECS) department for this opportunity. During my time teaching I worked closely with Dr. Kulkarni, Dr. Miller, and Dr. Gibbons. Each of them was eager to share their knowledge and experience teaching, which I attribute to my own teaching success, as evidenced by my receipt of the Paul F. Huebner Memorial award.

I worked on an applied research project with Dr. Rodolfo Torres and Dr. Joshua Rosen-

bloom, they are a pleasure to work with and I would especially like to thank them for their support over the past couple years. Finally, I'd like to give thanks to my lab mates Chao, Meenakshi, Xiaoli, as well as Dr. Alexios Koutsoukas for the discussion and camaraderie. And last but not least, I'd like to thank my wonderful wife Laura for her unending support over the last several years.

Abstract

Many important machine learning and data mining algorithms rely on a measure to provide a notion of distance or dissimilarity. Naive metrics such as the Euclidean distance are incapable of leveraging task-specific information, and consider all features as equal. A learned distance metric can become much more effective by honing in on structure specific to a task. Additionally, it is often extremely desirable for a metric to be sparse, as this vastly increases the ability to interpret or explain the measures produced by the distance metric. In this dissertation, we explore several current problems in distance metric learning and put forth solutions which make use of *structured sparsity*.

The contributions of this dissertation may be broadly divided into two portions. In the first portion (chapter 3) we begin with a classic approach in distance metric learning and address a scenario where distance metric learning is typically inapplicable, i.e., the case of learning on heterogeneous data in a high-dimensional input space. We construct a projection-free distance metric learning algorithm which utilizes structured sparse updates and successfully demonstrate its application to learn a metric with over a billion parameters.

The second portion (chapters 4 & 5) of this dissertation focuses on a new and intriguing regression-based approach to distance metric learning. Under this regression approach there are two sets of parameters to learn; those which parameterize the metric, and those defining the so-called "virtual points". We begin with an exploration of the metric parameterization and develop a structured sparse approach to robustify the metric to noisy, corrupted, or irrelevant data. We then

focus on the virtual points and develop a new method for learning the metric and constraints together in a simultaneous manner. We demonstrate through empirical means that our approach results in a distance metric which is much more effective than the current state of-the-art.

Machine learning algorithms have recently become ingrained in an incredibly diverse amount of technology. The primary focus of this dissertation is to develop more effective techniques to learn a distance metric. We believe that this work has the potential for rather broad-reaching impacts, as learning a more effective metric typically results in more accurate metric-based machine learning algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

A fundamental challenge in machine learning and data mining is finding a good representation of the data, one which reflects the inherent similarities and differences within the data. For example, considering text data, which is natively represented as a sequence of letters, the words "real" and "genuine" do not share a common sequence of letters, and could be considered quiet different. The words "similar" and "dissimilar" share a long common sequence and may be considered nearly identical. However, someone fluent in the English language would state that actually the opposite conclusion is correct.

Some insight into this process may be gained by looking into what is perhaps the most sophisticated of all computers, the human brain. A recent study [20] proposes that the visual cortex does not recognize scenes or images based directly on retinal cell activations, but rather it constructs richer structures (e.g., lines, textures, curves) where object recognition becomes possible. The conclusion is that a naive approach to measuring (dis)similarity (i.e., distance) can result in drastically incorrect decisions. It is often more insightful to consider the available structures or feature relationships in the data.

In distance metric learning, we are typically presented with a notion of similarity (or dissimilarity) and the task is to wade through a myriad of features determining which are useful in constructing a useful representation of the data. It is often the case that many features are irrelevant and only a few "key factors" are responsible for the differences. For example, we consider copies of the same image, each one rotated by a different amount. Comparing the images on each native degree of freedom (i.e., pixel by pixel) will indeed show that all the images are quite different, but not in a way which reflects the rotational differences between images. However, if we had a metric which were to discover and reflect the rotation as part of the learned distance, we would have an intuitive representation of these differences.

When making predictions from data, an effective distance metric learned on that data can make all the difference. Many important and widely-used machine learning algorithms rely

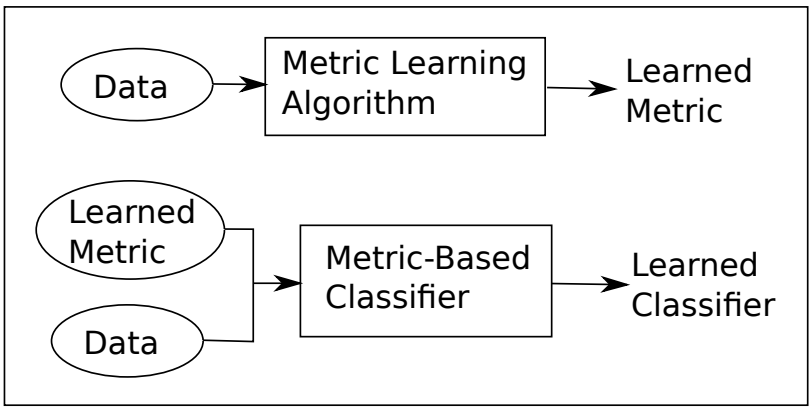

Figure 1.1: Relationship between a learned metric and a learned classifier.

on a metric to relay distances between instances (i.e., data samples) (Figure 1.1). A handful of examples include K-means clustering [54, 79], ranking [51, 57], K-Nearest-Neighbors [77], and Support Vector Machines [15]. An effectively learned metric can have a rather drastic impact on a learning task in terms of both performance and interpretability.

1.2 Motivating Examples

This section provides examples from real-world scenarios which serve as particularly good motivating examples for the work presented in this dissertation.

1.2.1 Chemical Toxicity Prediction

Year after year, a large number of untested synthetic chemicals are integrated into consumer products and become available on the market. As of now there are over 33 millions chemicals registered in the Chemical Abstract Service (CAS) database. While many chemical are safe, others are quite toxic to human health and may cause a number of life-threatening conditions [8]. The large number of new chemicals introduced each year make the prospect of testing each of them *in-vivo* (i.e., tested in animals) unrealistic. Instead, researchers have begun to experiment with *in-silico* techniques, that is, employing computational methods to predict toxic effects based on chemical structures. We envision metric learning techniques which

are capable of ranking chemicals based on their similarity to known toxic or safe chemicals. This will allow manufacturers to select chemicals which are most likely to be safe and avoid those most likely to be harmful. When using a metric which admits a sparse representation (i.e., it only relies on a small fraction of the available features), it may be possible to inspect the components most influencing the toxic effects, and avoid those in the design of future chemicals.

1.2.2 Multimedia Information Retrieval

The purpose of an information retrieval system is to return a list of items when given a query item. The items may be in the form of text, images, sound clips, or otherwise. The results are returned ranked by their similarity (i.e., dissimilarity/distance) with the query object. A motivating example is to identify an unknown person from a single image of their face. Considering the entire image may not be useful, as pixel values change drastically depending on occlusions, background, and lighting variations. In this case, we might wish to learn a metric which can selectively discard irrelevant information and hone in on key indicators such as skin tone, eye color, etc.

1.2.3 Social Network Analysis

Social networks have become a major part of peoples' lives and have grown particularly large, making it difficult for users to find new connections. The reasons for users to be connected (i.e., be similar) may be due to a number of different reasons. For example, a user may have connections with work colleagues due to shared professional interests, yet connections to friends may be due to shared interests in particular leisure activities. In other cases, reasons for similarity may be more mysterious (e.g., connections with family members). We see that the ideal representation (that which best explains the connections and similarities) may change depending on the user and the connections. This is in fact a key motivating factor for the contribution of Chapter 3 where we introduce a new method for learning a

local metric using sparse representations.

1.2.4 Theoretical Motivations

As we enter the era of "big data", there are several theoretical motivations to consider when learning a metric which admits a sparse representation of the data. A phenomenon called *concentration of distances* pertains to the inclination of all distances to become equal as the dimension of the feature space increases [7]. The variability in the distance between samples of different similarities is crucial for a metric to be effective. This motivates the use of metrics which have a low rank (i.e., the representation lives in a low*er*-dimensional space). Constraining a metric to be low rank is a problem known to be NP-hard and cannot be solved efficiently. Another option is to indirectly encourage the metric to be low rank by promoting sparsity in the metric. Inducing sparsity allows the metric to pick out a small number of "key factors" which dominate the similarity measure and has the advantage of good interpretability.

1.3 Contributions

This dissertation explores the case of distance metric learning with structured sparsity. Specifically, we propose three new algorithms for distance metric learning, each of which is tested under the setting of *supervised learning*. While related, each algorithm addresses a specific weakness in the state of-the-art and through a specific application of structured sparsity drives the learning of the metric in a new manner.

The first algorithm, *Sparse Compositional Local Metric Learning* (SCLM), is built to handle the case of learning from high-dimensional, heterogeneous data. The heterogeneous nature of the data implies that it may exhibit multiple modes, meaning the key factors or interactions may be non-uniform across the input space. Typically, local metric learning is applied to heterogeneous data, however, the computational cost of these methods make

their application to high-dimensional data extremely difficult. In our SCLM approach, we build upon the popular large-margin approach to metric learning, and are able to extend its reach to high-dimensional data through use of carefully designed update steps (each of which exhibits structured sparsity), which ensure that no expensive projections are ever needed. In order to handle the heterogeneous nature of the data, we propose a compositional metric, one which consists of both local and global components. Each metric is composed of a local component, and all share a common global counterpart. This approach allows a balance between local adaptivity and global consistency.

The second algorithm, *Robust Regressive Virtual Metric Learning with Structured Sparsity* (RRVMLSS), builds upon the recently proposed regressive virtual metric learning (RVML) [59]. RVML is a new approach to learning a metric which reduces the number of constraints from $O(n^2)$ to $O(n)$, where n is the sample size. We consider the case of learning with data containing noisy or irrelevant features. A common approach to learning a distance metric with noisy data is to regularize the metric to be low-rank; however, interpretability of the metric may suffer as sparseness is not guaranteed. We take the approach of introducing input sparsity to the metric, which may be done by learning the metric indirectly ($\boldsymbol{L}$, where $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$) and regularizing $\boldsymbol{L}$ with structured sparsity. This structured sparse design of $\boldsymbol{L}$ results in a sparse distance metric $\boldsymbol{M}$. We show via experimental means the performance of the algorithm on datasets from varying domains, and demonstrate the ability to recover the relevant features in a scenario with noisy data.

Finally, the third method is called *Regressive Virtual Metric Learning with Dynamic Margins* (RVML-DM). In our previous work with RRVMLSS we observed that the virtual points have a rather large influence on the quality of the distance metric. The regression-based approach to distance metric learning is relatively new and what characteristics are beneficial in virtual points is a topic not yet studied. We consider two aspects in the virtual points; (1) the discrimination potential of the virtual points, and (2) the compatibility of the virtual points with the distance metric. In our proposed method, we integrate these

considerations under a single multi-convex objective which eliminates the common two-step approach (i.e. constraint construction and metric optimization) to learning a distance metric. We show experimentally via a classification scenario that this algorithm produces competitive results with state of the art regressive virtual metric learning. Additionally, we produce several intriguing visualizations which provide insight into the success of our approach. Finally, we show that our method may be interpreted under the classic large-margin approach as learning with dynamic margin adjustments.

1.4 Dissertation Organization

The current chapter is given as an introduction to this work, with the remainder of the dissertation organized as follows.

Chapter 2 provides the necessary preliminaries to the dissertation and details the background of seminal and recent works related to distance metric learning. This is followed by a more detailed introduction to the primary contributions of this work. Chapters 3, 4, and 5 form the body of work on distance metric learning and detail the major contributions of this dissertation. Each of these chapters highlights a separate contribution to the field of distance metric learning. The chapters are written such that each stands independently from the others and may be read in any order. Finally, chapter 6 summarizes the dissertation with a conclusion and outlines several potential directions for future work.

Chapter 2

Background

2.1 Overview

This chapter provides some mathematical preliminaries and background information which are essential for a general understanding of the contributions of this work. After introduction of the mathematical preliminaries, we provide a background consisting of important works in metric learning and highlight some of the current challenges central to distance metric learning. After a more comprehensive background to distance metric learning has been introduced, this chapter introduces the contributions and further chapters of this dissertation in a more detailed manner.

2.2 The Definition of a Metric

Metric learning is a rich and varied topic, in this dissertation we are concerned specifically with *distance metric learning*, also sometimes referred to as dissimilarity function learning. For a function to be a valid metric (i.e., distance function) a number of properties must hold.

Definition A metric [63] is a mapping $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ over a vector space $\mathcal{X}$ such that for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X}$ the following properties hold:

1. $d(\boldsymbol{x}, \boldsymbol{y}) \leq d(\boldsymbol{y}, \boldsymbol{z}) + d(\boldsymbol{x}, \boldsymbol{z})$ (triangle inequality)
2. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non-negativity)
3. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{z})$ (symmetry)
4. $d(\boldsymbol{x}, \boldsymbol{y}) = 0 \implies \boldsymbol{x} = \boldsymbol{y}$ (uniqueness)

There are some distance/dissimilarity functions which do not satisfy the uniqueness property, in this case the metric is actually a *pseudo-metric*. However, in the machine learning and data mining literature pseudo-metrics are almost always referred to as metrics, and we shall do the same in this dissertation. It may also be useful to note that there is a body of similar work which focuses on learning a *similarity function* [4, 61, 60, 85], these works

learn a mapping from $\mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ and are typically a generalization of the cosine similarity function, therefore they do not normally satisfy the non-negativity or uniqueness properties.

The Mahalanobis[1] distance function (equation 2.1) forms the foundation of most works in distance metric learning. In the Mahalanobis distance, the matrix $\boldsymbol{M} = \boldsymbol{S}^{-1}$, where $\boldsymbol{S}$ is the covariance matrix of the data. By this definition all variables are defined and there is no learning to be done. However, virtually all works in *Mahalanobis distance metric learning* abuse this definition, $\boldsymbol{M}$ is treated as a parameterization of the distance and the goal is to learn a positive semi-definite (equation 2.2) parameterization of the metric ($\boldsymbol{M} \in \mathcal{S}_+$) satisfying some constraints.

Definition Mahalanobis Distance[2]

$$d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y})} \tag{2.1}$$

Definition A symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is positive semidefinite is for any vector $\boldsymbol{x}$ the follows holds true:

$$\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \geq 0 \tag{2.2}$$

2.3 Learning a Distance Metric

2.3.1 Distance Metric Learning Paradigms

Distance metric learning typically occurs under one of the three following learning paradigms:

- *Fully Supervised Learning*[3]*:* The algorithm has access to both the data samples $\boldsymbol{X} \in \mathcal{X}$ and the labels $\boldsymbol{y} \in \mathcal{Y}$. The labels are typically used to generate side-information in the form of neighborhood constraints which are used to guide the learning process. The quality of the side-information is typically never questioned.

[1]The term Mahalanobis comes from Mahalanobis of [55]

[2]Note that when $\boldsymbol{M}$ is the identity matrix, then we have the Euclidean distance.

[3]The fully supervised learning scenario is the easiest of the three distance metric learning paradigms in terms of evaluation. This could explain why it is so popular in the literature.

- *Weakly Supervised Learning:* The algorithm has access to the data samples $\boldsymbol{X} \in \mathcal{X}$ but has no access to the labels. The supervision comes in the form of similarity / dissimilarity constraints. For document classification this weak supervision could consist of citation links, or in the case of social networks the supervision could be in the form of graph connections (i.e. friend links). In this paradigm the constraints do not explicitly communicate class label information, but rather show which samples are most likely to be similar / dissimilar. In [5] it is suggested to think of this paradigm as having labels at the constraint level, rather than at the class level.

- *Semi-Supervise Learning:* In this paradigm the algorithm has access to labels or side information for only a small portion of the data. Algorithms of this type are often more resilient to overfitting than when training only on the labeled portion of the data.

2.3.2 Learning with Constraints

2.3.3 Pairwise Constraints

The simplest type of side information are pairwise constraints (equation 2.3), these specify what samples are "must-link" and which are "cannot-link" and are sometimes refereed to as "positive" and "negative" pairs. These constraints encourage the metric to pull similar points (set $\mathcal{S}$) together, while pushing dissimilar points (set $\mathcal{D}$) apart (see Figure 2.1).

$$\begin{aligned} \mathcal{S} &= \{(\boldsymbol{x}, \boldsymbol{x}) : \boldsymbol{x}, \boldsymbol{y} \quad \text{should be similar, "cannot-link"}\} \\ \mathcal{D} &= \{(\boldsymbol{x}, \boldsymbol{z}) : \boldsymbol{x}, \boldsymbol{z} \quad \text{should be dissimilar, "must-link"}\} \end{aligned} \tag{2.3}$$

2.3.4 Triplet Constraints

An alternative to pairwise constraints are to form constraints in the form of "triplets" or relative distance constraints (equation 2.4). Relative distance constraints are composed of triplets of the form $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, where $(\boldsymbol{x}, \boldsymbol{y})$ are encouraged to move together and $(\boldsymbol{x}, \boldsymbol{z})$ are

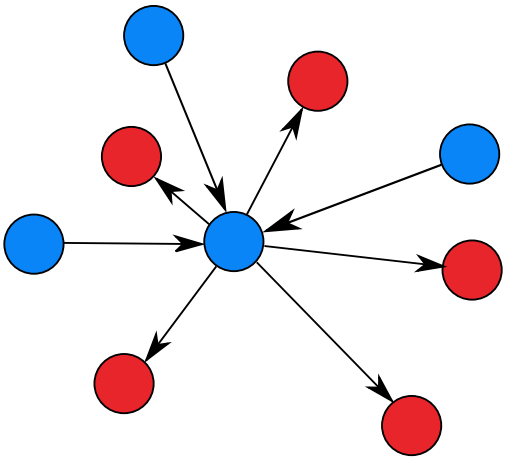

Figure 2.1: Visualization of pairwise constraints for the center point of the figure, similar points are pulled in, while dissimilar points are pushed away.

encouraged to move apart. A margin m is introduced to encourage a separation between $\{d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y})$ and $\{d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{z})$ (see Figure 2.2).

$$\mathcal{T} = \{d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y}) \text{ should be more similar than } d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{z}) \text{ by some margin } m\} \tag{2.4}$$

2.3.5 Virtual Points

A relatively new method called Regressive Virtual Metric Learning (RVML) was recently proposed by [59]. The key concept in RVML is the replacement of the pairwise or relative distance constraints with "virtual points". In lieu of instances/samples attracting or repelling one another, each is instead pulled towards a predetermined virtual point.

The main advantage this technique has over the classic relative distance based approaches is that the number of constraints are linear in the number of samples (rather than quadratic). RVML is an extremely new approach and to our knowledge the sole work of this type if [59]. We observe that in [59] a couple of approaches to finding the virtual points are proposed. It is still very much an open questions as to what means are best to find or learn the virtual

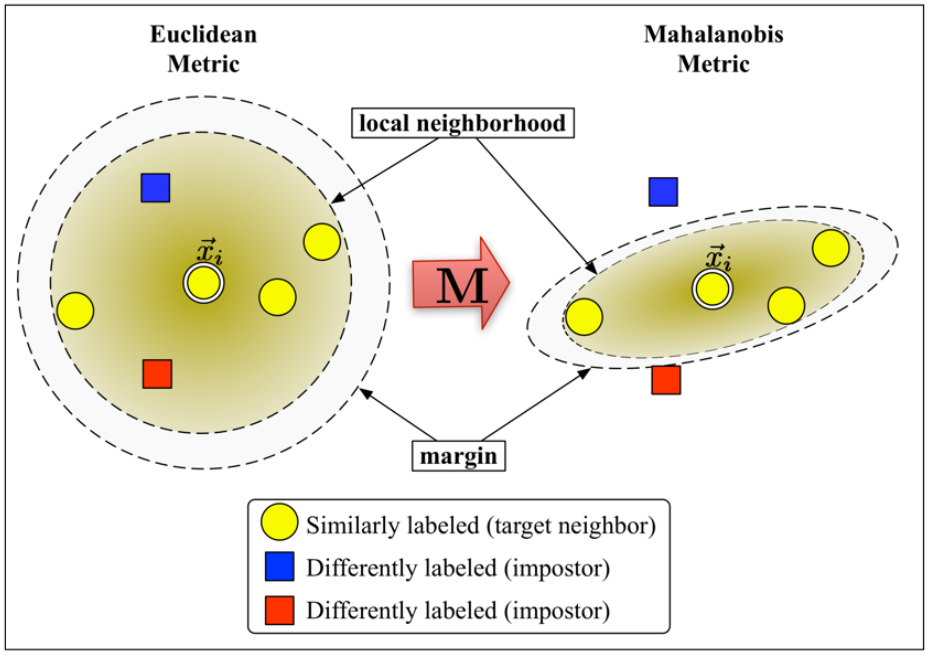

Figure 2.2: Visualization of large-margin nearest neighbors approach. Targets are pulled inwards and imposters pushed outwards such that distances are separated by at least the margin amount. (Image used under the Creative Commons 3.0 license.)

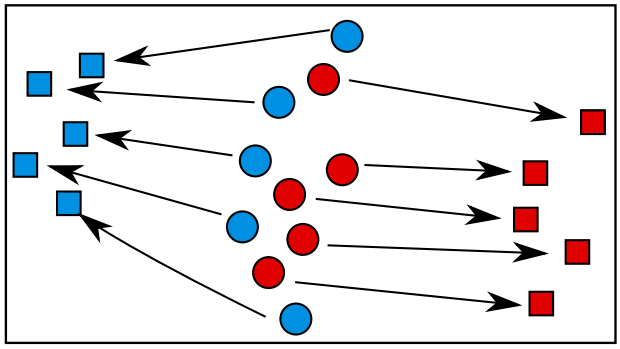

Figure 2.3: Visualization of Regressive Virtual Metric Learning approach, where each instance (circle) is pulled towards a corresponding virtual point (square). Colors represent instances having different class labels.

points.

2.4 Direct and Indirect Distance Metric Learning

2.4.1 Direct Distance Metric Learning

A straightforward approach to learning the distance metric is to learn $\boldsymbol{M}$ in a direct manner. Many methods (e.g. [25, 79, 88]) take this approach. A major disadvantage of these methods is that $\boldsymbol{M}$ needs to be projected onto the positive semidefinite cone (often at every iteration) in order to ensure that it is positive semi-definite. This is a costly computation with cubic compexity, methods which learn $\boldsymbol{M}$ directly are typically only applicable on relatively small datsets.

2.4.2 Indirect Distance Metric Learning

It is possible to learn the metric parameters $\boldsymbol{M}$ in an indirect manner through the Cholesky decomposition $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$. This has the advantage that $\boldsymbol{M}$ becomes positive semidefinite by construction and eliminates any need to calculate a projection to keep $\boldsymbol{M}$ positive semidefinite. The Cholesky decomposition[4] of a positive definite matrix is unique [27]. This form of decomposition of $\boldsymbol{M}$ elucidates the following simple connection between learning a distance metric and learning a representation.

Lemma 2.4.1. *Given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$, learning a Mahalanobis distance $d_{\boldsymbol{M}} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^+$ is equivalent to learning an embedding function parameterized by $\boldsymbol{L} \in \mathbb{R}^{q \times p}$, where $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{M}$.*

$$
\begin{aligned}
d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y}) &= (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y}) \\
&= (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{L}^T \boldsymbol{L} (\boldsymbol{x} - \boldsymbol{y}) \\
&= (\boldsymbol{L}(\boldsymbol{x} - \boldsymbol{y}))^T \boldsymbol{L} (\boldsymbol{x} - \boldsymbol{y}) \\
&= (\boldsymbol{L}\boldsymbol{x} - \boldsymbol{L}\boldsymbol{y})^T (\boldsymbol{L}\boldsymbol{x} - \boldsymbol{L}\boldsymbol{y})
\end{aligned}
$$

[4] $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$

2.5 Current Challenges in Distance Metric Learning

2.5.1 Input Space Dimension – Scalability

Given a dataset $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, where p denotes the number of features. The dimension of the input space (p) presents a major challenge in terms of scalability when learning a Mahalanobis distance function. As mentioned previously, to maintain the properties of a distance metric the matrix $\boldsymbol{M} \in \mathbb{S}_+^p$ must be positive-semidefinite. A common approach to maintaining that $\boldsymbol{M} \in \mathbb{S}_+^p$ is to alternatively update $\boldsymbol{M}$ and project it onto the feasible set. This projection involves a full eigendecomposition of $\boldsymbol{M}$, the negative eigenvalues are set to zero and $\boldsymbol{M}$ is reconstructed[5]. It is well known that a full eigendecomposition is computationally expensive and costs on the order of $O(p^3)$. This expense severely inhibits the application of learning $\boldsymbol{M}$ for datasets with even a moderate number of features.

2.5.2 Input Space Dimension – Generalization

In addition to the scalability issues, the generalization ability of the model becomes a major concern due to the particularly large number of variables to be fit. The number of parameters grows on the order of $O(p^2)$, meaning that for a problems with as few as 1,000 features, roughly 500,000 parameters must be learned! As we push towards "big-data" scenarios, simply storing the full matrix $\boldsymbol{M}$ may become infeasible. Due to the large number of features there is a very real tendency to overfit the data. In machine learning we typically apply some kind of regularization to limit the complexity of the model. Two options are available for distance metric learning which are often referred to as *input sparsity* and *output sparsity*.

Output sparsity refers to limiting the complexity of the model in the output space. This is typically accomplished by encouraging the metric matrix $\boldsymbol{M}$ to be low-rank (i.e. the projection dimension of $\boldsymbol{L}$ is low-dimensional). Some works have found some success using different types of regularization such as manifold regularization [93], trace norm regulariza-

[5]This process is often referred to as projecting $\boldsymbol{M}$ onto the positive semi-definite cone.

tion [84], and capped trace norm regularization [34]. Unfortunately, these methods are not very efficient as optimizing $\boldsymbol{M}$ subject to a low rank constraint is a problem known to be NP-hard [5].

Input sparsity refers to the application of regularization on the input space of the metric and has a similar idea to that of applying a lasso penalty to a linear regression function. Unlike linear regression, sparsity-inducing regularization cannot be naively applied to the metric, as this could destroy the positive semidefinite property of $\boldsymbol{M}$. One option is to apply group regularization to symmetric entries of $\boldsymbol{M}$; however, under this approach the number of penalty groups quickly grows to an unreasonable size. A recent work [52] in similarity metric learning has proposed a clever update scheme guaranteeing sparseness, but this approach has not yet been applied to distance metric learning.

2.5.3 Sample Size

Construction of the constraints is a crucial factor in learning an effective metric. Current techniques rely on either pairwise or triplet(relative distance) constraints which grow at a rate of $O(n^2)$ or $O(n^3)$ respectively. A common practice in dealing with the constraints is to only consider a select subset of the total constraint set. This is accomplished by only considering a select number of constraints for each point, the constraints preserved are between each point and its closest neighbors as determined by the Euclidean distance. When constructing the constraint set the distance between all pairs of points must be calculated. This has a rather expensive computational complexity of $O(n^2)$.

2.5.4 Legitimacy of Constraints

Save for one work that we are aware of [33], the legitimacy of the pairwise and triplet constraints are virtually never questioned. We believe this could lead to a situation where a few largely violated constraints could poison the metric learning process. Another pertinent consideration in the constraint construction process is that it is entirely isolated from the

distance metric learning procedure. It is well known that the performance of a metric learned in this manner is particularly sensitive to the ability of the Euclidean distance to select good target neighbors[6] [5]. Our viewpoint on this is that the Euclidean distance is in some sense a hyper-parameter of the distance metric learning algorithm. To our knowledge other distance metrics have not been experimented with in selecting target neighbors for constraint construction.

From a theoretical perspective, the concentration of distances phenomenon states that as the dimension of the feature space increases, all distances concentrate [7]. We can expect that as we move to high-dimensional input spaces, the constraints may become less and less meaningful. Despite these shortcomings, there are virtually no works emphasizing how to learn or construct the constraints beyond the status quo.

2.6 Outline and Summary of Contributions

In the remainder of this dissertation, we develop several solutions based on the use of *structured sparsity* to address issue with state of the art distance metric learning. Chapter 3 begins with the classic large-margin distance metric learning approach and addresses the specific case of learning a local distance metric in a high-dimensional feature space. The remaining two works innovate upon the newly introduced regression-based approach to distance metric learning. Chapter 4 introduces a mechanism which makes the regressive virtual metric learning approach robust to noisy or irrelevant inputs and results in a structured sparse metric. Chapter 5 considers the question of what characteristics are desirable in the virtual points. We discuss two aspects we believe to be in important for learning virtual points, and propose an entirely new metric learning approach which may be interpreted as sparse discriminative representation learning with structured sparsity.

[6]Target neighbors: those points used to form constraints.

2.6.1 Sparse Compositional Local Metric Learning

In chapter 3, we consider the case of learning a local distance metric for heterogeneous high-dimensional data. The typical approache [19, 50, 74, 77, 88] to learning a distance metric in a high-dimensional feature space is to first compress the data via a projection onto a low-dimensional manifold. Projections of this type are typically computationally expensive operations and are often sensitive to unit changes or rescalings of the data (e.g. Principal Components Analysis).

Some works address this problems through the use of a low-rank (structured) approach [26, 43, 77]. These works operate by indirectly learning a matrix $\boldsymbol{L}$ such that $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$. This removes the need to enforce that $\boldsymbol{M}$ is positive semi-definite, and the rank of $\boldsymbol{L}$ can be controlled by selecting the trailing dimension of $\boldsymbol{L}$. However, $\boldsymbol{L}$ is typically full rank and and may be challenging to store. Finally, works of this type often results in a non-convex objective and are often plagued by many local minima [45].

The few works which do operate on a high dimensional feature space [21, 52, 53, 24] all consists of global metrics, meaning that they assume that the feature interactions are relatively constant across the feature space. Heterogeneous data can exhibit multiple modes which current methods are unable to appropriately handle. A popular approach is to focus on learning multiple metrics, each of which is local to a different spacial area of the input space. Works of this type have the ability to capture heterogeneous feature interactions, and have in many cases been shown in mayn cases capable of outperforming their global counterparts [21, 23, 32, 74, 77, 91]. However, these works on local metric learning work with a full matrix $\boldsymbol{M}$ and are unable to be applied to high-dimensional data.

Our aim in chapter 3 is to learn a distance metric for heterogeneous data directly in a very high dimensional feature space. Our proposed method features many desirable characteristics in a distance metric learning algorithm, including:

- Learns a high-dimensional $\boldsymbol{M}$ directly in the feature space.

- Structured sparse updates to $\boldsymbol{M}$ ensure feasibility at every point of the optimization (projection-free).
- $\boldsymbol{M}$ is compositional and contains global and local metric components, allowing it to balance local adaptability and global consistency.
- The rank of $\boldsymbol{M}$ can be controlled by early termination of the optimization algorithm.

2.6.2 Robust Regressive Virtual Metric Learning with Structured Sparsity

Regressive Virtual Metric Learning (RVML) [59] is a new approach to distance metric learning based on regression. The advantage of RVML is that it only requires a linear number of constraints and $\boldsymbol{M}$ is learned in a projection-free manner. However, RVML has no mechanism to discard or reduce the influence of noisy, corrupted, or irrelevant features.

Our aim in chapter 4 is to introduce a more robust version of RVML so that it is able to reject (or reduce the influence of) noisy and/or irrelevant features. We propose a new distance metric learning method based on RVML which utilizes structured sparsity to increase the robustness of RVML. In our approach we learn the metric indirectly and carefully design the structure of the penalty such that $\boldsymbol{L}$ is row sparse. The consequence of this is that $\boldsymbol{M}$ becomes sparse as well, which makes the distance metric more interpretable.

2.6.3 Regressive Virtual Metric Learning with Dynamic Margins

In the work of chapter 4 we observed that the rank of $\boldsymbol{M}$ is controlled by the dimension of the virtual points. The only previous work on regression-based distance metric learning suggests two rudimentary methods for constructing the virtual points, neither of which are able to control the rank of $\boldsymbol{M}$. This leads us to the question; what characteristics are desirable in the virtual points and what is a good means to solve for them?

In chapter 5, we propose two ideas which may lead to good virtual points; 1) potential discriminatory ability of the virtual points, and 2) the compatibility between the virtual points and the form of the metric. We design an approach based on these idea and reveal connections between the proposed approach and dictionary learning and sparse coding [2]. Continuing on this thread we move to show that the proposed approach of chapter 5 may be considered under the relative distance metric learning approach as dynamic adjustment of the margins during the learning process.

Chapter 3

Sparse Compositional Local Metric Learning

3.1 Introduction

Distance metrics are the core of many machine learning algorithms, including k-Means clustering[79], ranking[57], k-Nearest Neighbors[77] and many others. In this chapter, we address the problem of learning a locally-adaptive distance metric for data with high dimensional input (i.e. feature) spaces. Specifically, we propose a method for local distance metric learning which learns the matrix parameterizing the metric directly in the input space. A distinguishing property of our proposed method is that it is both *locally adaptive* and *scalable* with respect to the dimension of the input space.

Learning a distance metric is a well-studied problem, refer to the surveys [5] and [45] for a summary of recent works. In general, learning a distance metric is a computationally expensive procedure, with most existing algorithms having from $O(p^2)$ to $O(p^{6.5})$ computational complexity, where p is the dimension of the input space. This computational expense stems from the requirement that the metric matrix $\boldsymbol{M} \in \mathbb{R}^{p \times p}$ must be symmetric and positive semi-definite (p.s.d). The most common approach to maintain that $\boldsymbol{M}$ is p.s.d is via a projection onto the positive semi-definite cone after each iteration, with a computational cost on the order of $O(p^3)$. In addition, as p is scaled upwards it may simply become infeasible to store $\boldsymbol{M}$ in memory. Finally, learning on the order of $O(p^2)$ parameters greatly increases the chance of over-fitting.

In light of the above challenges, relatively little work has been done on learning $\boldsymbol{M}$ directly in the high dimensional feature space [21, 52, 53, 24]. Until recently, most work on distance metric learning has addressed high dimensional input spaces by first compressing the space via a projection onto a low-dimensional manifold[19, 50, 74, 77, 88]. This is typically done via an eigen-decomposition, the computational cost of which is obviously prohibitive. Additionally, projections of this nature are sensitive to unit changes or re-scaling of the data, and may not preserve information which is well-suited to learning a distance from. This is in contrast to our work, which does not compress or reduce the input space dimension via a projection or other means.

A number of works address the case of metric learning for high-dimensional data by taking a low-rank(structured) approach [26, 43, 77]. These works operate by indirectly learning an $\boldsymbol{L}$ such that $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$, this removes the need to enforce that $\boldsymbol{M}$ is positive semi-definite, and the rank of $\boldsymbol{L}$ can be controlled by selecting the trailing dimension of $\boldsymbol{L}$. This reduces the total number of independent parameters to be learned, but also introduces some drawbacks. The matrix $\boldsymbol{L}$ is typically full rank and may have challenging storage requirements. Additionally, virtually all works of this type result in a non-convex objective, which may be difficult to optimize and is often plagued with many local minima [45]. One work has resulted in a convex objective[18], but limits the solution space to the span of $\boldsymbol{L}$. These works also have no means to control the sparsity of $\boldsymbol{M}$, which may harm the interpretability of the model. Our proposed method is convex, has no restrictions introduced by the range of a low-rank projection operator, and encourages interpretability through the use of sparsity inducing penalties.

Traditional approaches to metric learning take a global approach, and make the assumption that feature interactions are consistent across the input space. In cases where the decision boundary is too complex, or the data is multi-modal, a global metric may be too inflexible. A popular approach is to focus on learning multiple metrics, each of which is local to a different spacial area of the input space. Works of this type have the ability to capture non-heterogeneous feature interactions, and have in many cases been show capable of outperforming their global counterparts [21, 23, 32, 74, 77, 91]. In our work we propose the use of a two part compositional metric consisting of both global and local components. Our approach is to separate portions which model global data trends, and those interactions which are confined to a local area. We structure the metric such that the global portion is limited to the diagonal, and the local portion is sparse and positive semi-definite.

Specifically, we introduce a sparse compositional metric for high-dimensional data which is locally adaptive. We propose a two part compositional metric, allowing it to capture global trends, while remaining sensitive to local interactions. We develop an efficient Frank-Wolfe

style alternating optimization algorithm which maintains that $\boldsymbol{M}$ remains within the constraint region between all iterates. This allows us to avoid any expensive projections onto the positive semidefinite cone, enforces our choice of structure on the global portion of the metric, and removes the requirements of ever having to store a full matrix $\boldsymbol{M}$. Finally, because of the compositional design of our algorithm, it lends well to an efficient implementation.

In summary, in this chapter we claim the following contributions:

- We propose a new algorithm for learning a sparse compositional metric for high-dimensional data, which consists of both global and local components.
- To our knowledge, our proposed method is the first locally-adaptive distance metric which is learned directly in the input space.
- We provide an empirical evaluation of our method against the current state of the art via a classification scenario.
- Finally, we make our code freely available for download, to aid in research reproducability.

3.2 Background and Related Work

3.2.1 Frank-Wolfe Style Optimization

Frank-Wolfe optimization (also referred to as the conditional gradient method) [22] has recently experienced a resurgence of interest, primarily due to its capability of producing projection-free updates. Much recent work has focused on variants of Frank-Wolfe methods and their convergence properties ([36, 37, 46] and others), with numerous applications including video co-localization [40], particle filtering [47], any many others. Updates in the Frank-Wolfe method are typically much cheaper in comparison to full projections, but often require many more iterations for convergence. Additionally, Frank-Wolfe suffers from sublinear convergence rates when the solution lies on the border of the constraint region [46].

Frank-Wolfe based techniques operate on problems of the form:

$$\min_{\boldsymbol{x} \in \mathcal{S}} f(\boldsymbol{x})$$

where f is a convex function with a smooth Lipshitz continuous gradient and $\mathcal{S}$ is a compact convex set. Each Frank-Wolfe update is calculated by linearizing f about the current iterate $\boldsymbol{x} \in \mathcal{S}$ and solving a linear subproblem for the descent direction. The solution to each subproblem is given by the linear minimization oracle (LMO) (equation 3.1), where $< \cdot, \cdot >$ denotes the inner Frobenius norm and $\nabla f(\boldsymbol{x})$ is the gradient of f at $\boldsymbol{x}$.

At any point during the Frank-Wolfe optimization procedure, the solution may be given by the combination $\alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2 + ... + \alpha_k \boldsymbol{x}_k$, where $\sum_{i=1}^{k} \alpha_i = 1$ and $\alpha_i > 0, \quad i \in \{1, ..., k\}$. Each $\boldsymbol{x}_i$ is a solution found by solving the LMO at some point in the optimization. Note that the vectors $\boldsymbol{x}_i$ are commonly referred to as "atoms". The power of this method lies in the fact that as long as $\mathcal{S}$ is convex and compact, the solution will never leave the constraint region.

$$\min_{\boldsymbol{x} \in \mathcal{S}} < \boldsymbol{x}, \nabla f(\boldsymbol{x}) > \tag{3.1}$$

Sparsity may be induced by the Frank-Wolfe technique in two ways:

1. During each iteration, a single atom is added to the solution. The total number of atoms is upper-bounded by the maximum number of iterations, the solution is then the result of a sparse combination of basis atoms.

2. In the case where each atom $\boldsymbol{x} \in \mathcal{S}$ itself is sparse, sparsity is produced as a combination of sparse basis atoms. In this situation the solution need not be stored directly, only the basis atoms and their corresponding weights are stored. This is the approach we take in this chapter.

Finally, the quality of the solution may be monitored via an upper bound on the duality gap (equation 3.2), this is often referred to as the Frank-Wolfe gap. In equation 3.2, $\boldsymbol{d}_{FW}$ denotes the Frank-Wolfe direction found by the linear minimization oracle.

$$\text{Frank-Wolfe Gap} = \quad < -\nabla f(\boldsymbol{x}), \boldsymbol{d}_{FW} > \tag{3.2}$$

3.2.2 Distance Metric Learning in High Dimensional Input Spaces

The majority of metric learning has been done under the framework of the Mahalanobis distance function (see the surveys [5, 45]). With many local metric learning works shown capable of outperforming their global counterparts [35, 39, 48, 64, 66, 80, 91, 92]. However, these works are not able to scale to very high dimensions. Until recently, very few works have concentrated on metric learning directly in high dimensional input spaces [21, 24, 52, 67]. We restrict our discussion of the related work only to those papers which focus on handling the case of high dimensional data.

The work of [21] learns a Mahalanobis distance metric from only dissimilarities, and shows that this problem is equivalent to learning a Support Vector Machine(SVM) with a

quadratic kernel. The other contribution of [21] is that of local invariance to transformations of the data. However, the type of transformation must be known *apriori* and only the case of rotational invariance for image data is studied.

In [24], online distance metric learning is considered in the scenario of multimedia content retrieval. They accelerate the optimization process by constraining the metric matrix $\boldsymbol{M}$ such that all off-diagonal matrix entries are equal to zero. This prevents the metric from learning any iterations between features, whether on a local or global level. It is our belief that this assumption is too restrictive, as large and complex data spaces may have many 2nd-order interactions which could be pertinent for performance.

The recent works [52, 88] also focus on the case of metric learning via a Frank-Wolfe style optimization procedure. The work of [88] is a sparse compositional metric learning technique. While similar in name to this work, the approach is quite different. They focus finding a sparse combination of basis functions with which to construct the overall metric matrix $\boldsymbol{M}$. Each basis function is found as the leading eigenvector of the gradient matrix, the expense of solving for the leading eigenvector may limit its scalability. The solution of [88] consists of a sparse combination of basis atoms (each possibly full), while in our proposed technique the solution is a composition of sparse atoms.

Finally, the work of [52] learns a similarity metric directly in the high dimensional feature space. The method of this chapter shares some similarities with this work in that both utilize Frank-Wolfe optimization techniques. Besides the obvious difference in the metric learned(similarity vs. Mahalanobis), there are several contrasts. In [52], a single global similarity is learned, and local information in the dataset is not considered. In our work, we learn a set of local metrics which share a global component. The objective of [52] is to solve a method in a high dimensional input space quickly, in this chapter we go beyond [52] through the addition of local adaptability.

3.3 Methodology

3.3.1 Overview

Given a subset of the sample space, we wish to learn a Mahalanobis distance metric of the form $d(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y})$, where $\boldsymbol{M} = \boldsymbol{M}_G + \boldsymbol{M}_L$ consists of a diagonal metric $\boldsymbol{M}_G$ capturing long range trends and $\boldsymbol{M}_L$ modeling local interactions between variables. We constrain both matrices $\boldsymbol{M}_G$ and $\boldsymbol{M}_L$ to be symmetric positive semi-definite $\boldsymbol{M_G}, \boldsymbol{M_L} \in \mathbb{S}_+^d$.

Following many previous works in metric learning [65, 77], we make use of relative distance constraints in the form of triplets (equation 3.3). In 3.3, $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ share the same label, and $\boldsymbol{z}_i$ has a different label.

$$\mathcal{T} = \{\boldsymbol{x}_i \text{ should be close to } \boldsymbol{y}_i \text{ than to } \boldsymbol{z}_i\}_{i=1}^T \tag{3.3}$$

For each point $\boldsymbol{x}_i$, a set of triplet constraints may be formed by finding the nearest neighbors to that point, and labeling points as either similar or dissimilar, depending on the similarity to the label of $\boldsymbol{x}_i$. Alternatively, in the case where there are too many samples to form a distance matrix, points can be sampled randomly to form constraints. We choose to take a large-margin approach, and constrain the distance between dissimilar points to be larger than the distance between similar points by some margin m_i. Once the triplet constraints are formed, we guide the learning process of $\boldsymbol{M}$ with constraint violations.

Given a Mahalanobis distance function parameterized by $\boldsymbol{M}$ $d_M(\cdot, \cdot)$, a set of triplet points

Table 3.1: Summary of Data Characteristics

Dataset	**# classes**	**# Features**	**# Samples**
CNAE-9	9	856	1080
BBC-Sports	5	4613	737
BBC-News	5	9635	2225
TDT2-30	30	36771	9394
Madelon	2	500	2600

$(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i)$ and the corresponding margin m_i, the constraint is violated when $d_M(\boldsymbol{x}_i, \boldsymbol{y}_i) + m_i > d_M(\boldsymbol{x}_i, \boldsymbol{z}_i)$. This may be represented as a hinge loss function to be minimized (equation 3.4).

$$[d_M(\boldsymbol{x}_i, \boldsymbol{y}_i) + m_i - d_M(\boldsymbol{x}_i, \boldsymbol{z}_i)]_+ \tag{3.4}$$

The hinge loss (equation 3.4) has a discontinuous gradient and is ineligible to be solved with first-order techniques. To alleviate this we approximate the hinge loss with a smooth version (equation 3.5). Note that equation 3.4 may accommodate different margin values by simply shifting the input to the smooth hinge loss function by some desired amount.

$$h(x) = \begin{cases} x - \frac{1}{2} & x \geq 1 \\ \frac{1}{2}x^2 & 0 < x < 1 \\ 0 & x \leq 0 \end{cases} \tag{3.5}$$

3.3.2 Sparse Compositional Local Metrics

Given L local areas, each of which contains a set of samples $\mathcal{X}_i, i \in \{1, ..., L\}$, the constraint set and associated margins may be constructed. The set of constraints associated with each sample set $\mathcal{X}_i$ is denoted as $\mathcal{C}_i$. Each element of $\mathcal{C}_i$ is a 4-tuple consisting of three samples $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ and the desired margin m. For local sample set $\mathcal{X}_i$, we learn a compositional metric which consists of an exclusive local component parameterized by the local metric matrix $\boldsymbol{M}_i \in \mathcal{S}_+^d$ and a shared component $\boldsymbol{M}_G \in \mathcal{S}_+^d$ which is constrained such that all non-diagonal elements are equal to zero.

Integrating the above information, the objective is formed to optimize over(equation 3.6). Sparsity is introduced in each metric through the use of L_1-regularization, with the variables $\lambda_1, ..., \lambda_L, \lambda_G$ forming the boundaries of the constraint region associated with each metric matrix. Note that in equation 3.6 the local and global portions of the metric have been separated, as $(\boldsymbol{x} - \boldsymbol{y})^T(\boldsymbol{M}_G + \boldsymbol{M}_i)(\boldsymbol{x} - \boldsymbol{y}) \equiv (\boldsymbol{x} - \boldsymbol{y})^T\boldsymbol{M}_G(\boldsymbol{x} - \boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y})^T\boldsymbol{M}_i(\boldsymbol{x} - \boldsymbol{y})$.

$$
\begin{aligned}
\text{min.} \quad & \sum_{l=1}^{L} \sum_{i=1}^{C_l} [d_{\boldsymbol{M}_G}(\boldsymbol{x}, \boldsymbol{y}) + d_{\boldsymbol{M}_l}(\boldsymbol{x}, \boldsymbol{y}) + m_i^l \\
& - d_{\boldsymbol{M}_G}(\boldsymbol{x}, \boldsymbol{z}) - d_{\boldsymbol{M}_l}(\boldsymbol{x}, \boldsymbol{z})]_+ \\
\text{s.t.} \quad & \boldsymbol{M}_G \in \mathcal{S}_+^d \\
& \boldsymbol{M}_G(i,j) = 0, \quad \forall i \not= j \\
& \boldsymbol{M}_l \in \mathcal{S}_+^d, \quad \forall l \in \{1, 2, ..., L\} \\
& ||\boldsymbol{M}_G||_1 < \lambda_G \\
& ||\boldsymbol{M}_l||_1 < \lambda_l, \quad \forall l \in \{1, 2, ..., L\}
\end{aligned}
\tag{3.6}
$$

3.3.3 Maintaining Feasibility of Iterates

A pair-wise variant of the Frank-Wolfe (a visualization of the Frank-Wolfe update is shown as Figure 3.1) procedure was selected to minimize equation 3.6. Besides removing the need for projection operators, the pair-wise variant only requires the update of two atoms per iterate, in contrast with vanilla Frank-Wolfe, which requires the weights of all atoms to be updated at every iteration. Our objective function (equation 3.6) is a function of L+1 matrices $(\boldsymbol{M}_1, ..., \boldsymbol{M}_l, \boldsymbol{M}_G)$. The function is convex in each matrix with respect to the other matrices. We optimize this using an alternating pairwise Frank-Wolfe method where the local matrices are updated, followed by the global metric. We leave the variables $\lambda_1, ..., \lambda_L, \lambda_G$ which determine the constraint region bounds as tunable parameters.

Projections are avoided by careful selection of the update directions and step sizes, ensuring that $\boldsymbol{M}_1, ..., \boldsymbol{M}_L$ never leaves the constraint region. We use the following rank-1 update matrices $\boldsymbol{P}^{(i,j)}, \boldsymbol{N}^{(i,j)}$ in updating $\boldsymbol{M}_1, ..., \boldsymbol{M}_L$ and restrict the updates of $\boldsymbol{M}_G$ to $\boldsymbol{P}_\lambda^{(i,j)}$ where $i = j$. These update types were first proposed by [36] and subsequently used in [52]. Updating the matrices in this manner allows us to maintain the positive semi-definiteness of $\boldsymbol{M}_1, ..., \boldsymbol{M}_l, \boldsymbol{M}_G$ while still producing a sparse iterate. This is key to the scalability of our algorithm with respect to the size of the feature space.

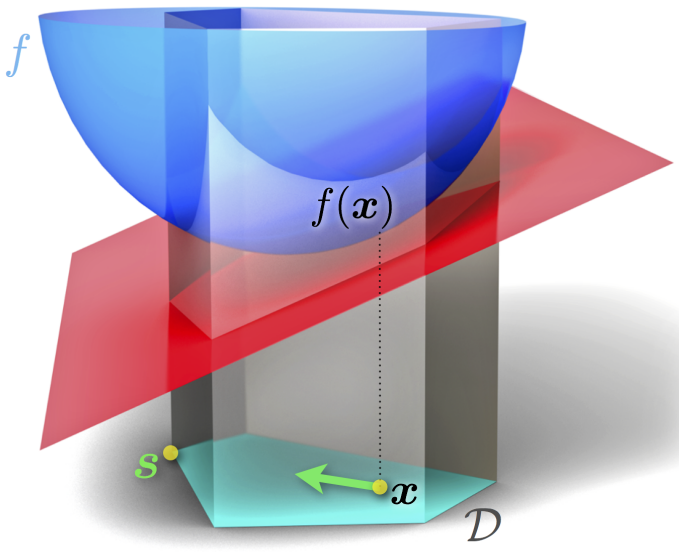

Figure 3.1: Visualization of Frank-Wolfe update procedure. $f(x)$ is the objective with x being the current iterate, s is a point on the simplex and $\mathcal{D}$ represents the constraint region. Credit to Stephanie Stutz for contribution to public domain, labels added by Martin Jaggi, CC BY 4.0, `https://commons.wikimedia.org/w/index.php?curid=35484532`

$$\boldsymbol{P}_{\lambda}^{(i,j)} = \lambda(\boldsymbol{e}_i + \boldsymbol{e}_j)(\boldsymbol{e}_i + \boldsymbol{e}_j)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\boldsymbol{N}_{\lambda}^{(i,j)} = \lambda(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & -\lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -\lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

The design of the update steps handles the constraint, leaving a smooth unconstrained objective function to minimize 3.7, where $h(\cdot)$ denotes the smooth hinge loss function 3.5.

The gradient for each $\boldsymbol{M}_i$ is calculated on a per-constraint basis, with each constraint violation contributing a direction and a scaling factor determined by the gradient of the smooth hinge loss function. This is shown in equations 3.8 and 3.9.

$$\begin{aligned} \text{min.} \quad & \sum_{l=1}^{L} \sum_{i=1}^{C_l} h(d_{\boldsymbol{M}_G}(\boldsymbol{x}, \boldsymbol{y}) + d_{\boldsymbol{M}_l}(\boldsymbol{x}, \boldsymbol{y}) + m_i^l \\ & - d_{\boldsymbol{M}_G}(\boldsymbol{x}, \boldsymbol{z}) - d_{\boldsymbol{M}_l}(\boldsymbol{x}, \boldsymbol{z})) \end{aligned} \tag{3.7}$$

$$g(f(x), \frac{\partial f(x)}{\partial x}) = \begin{cases} \frac{\partial f(x)}{\partial x} & x \geq 1 \\ f(x) \cdot \frac{\partial f(x)}{\partial x} & 0 < x < 1 \\ \mathbf{0} & x \leq 0 \end{cases} \tag{3.8}$$

$$\frac{\partial d_{\boldsymbol{M}_l}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})}{\partial \boldsymbol{M}_l} = (\boldsymbol{x} - \boldsymbol{y}) \cdot (\boldsymbol{x} - \boldsymbol{y})^T - (\boldsymbol{x} - \boldsymbol{z}) \cdot (\boldsymbol{x} - \boldsymbol{z})^T \tag{3.9}$$

3.3.4 Algorithm

Given the desired number of local metrics c, we begin by clustering the samples into c distinct groups. This may be accomplished via any clustering method, in our experiments we use either a sparse k-means approach or k-means clustering with the cosine distance. Constraints are formed for each point, this is done by taking the nearest neighbors which are "friends" (same label) and "impostors" (different label), then generating relative distance constraints in the form of triplets.

Given the metric matrices $\boldsymbol{M}_i, i \in \{1, .., c\}$ corresponding to each cluster and the global metric matrix $\boldsymbol{M}_g$, the task is to minimize the objective. The objective is convex with respect to each $\boldsymbol{M}_i$ when the others are held constant, allowing us to conduct the optimization in an alternating fashion. (Note that the local components $\boldsymbol{M}_1, ..., \boldsymbol{M}_c$ are independent when $\boldsymbol{M}_g$ is held constant and may be updated simultaneously if desired.)

The iterates $\boldsymbol{M}_1, ..., \boldsymbol{M}_c, \boldsymbol{M}_g$ must be initialized to lie within the feasible set. This is

accomplished by initializing each $\boldsymbol{M}_i$ to any valid atom in $\mathbb{S}_d^+$ and assigning it a weight of one. Note that the matrices $\boldsymbol{M}_1, ..., \boldsymbol{M}_c, \boldsymbol{M}_g$ are never formed explicitly, instead each matrix $\boldsymbol{M}_i$ is represented by a set of atoms $\mathcal{A}_i$ and atom weights $\mathcal{W}_i$ such that $\sum \mathcal{W}_i = 1$. This allows the addition, removal, and weight manipulation of individual atoms in a fast and efficient manner which pairs naturally with Frank-Wolfe style algorithms.

$$\begin{aligned} &\min_{\boldsymbol{S}} \langle \nabla f(\boldsymbol{M}_G), \boldsymbol{S} \rangle \\ &\boldsymbol{S}(i,j) = 0, \forall i \neq j \quad s.t. \quad \boldsymbol{S} \in \mathbb{S}_d^+ \end{aligned} \tag{3.10}$$

Once each $\boldsymbol{M}_i$ has been initialized, the algorithm proceeds by alternating through each local metric and the shared diagonal metric, updating each in turn. Each update consists of a calculation of the gradient, then solving the corresponding global or local linear minimization oracle (equation 3.10 & 3.11, respectively) for the Frank-Wolfe direction d_{FW}. In the pairwise variant of Frank-Wolfe, the best atom is identified from which to pivot weight away from. The advantage of this approach is that only two atoms must be manipulated for each iteration, this is in contrast to the full Frank-Wolfe method in which the weight of every atom in the active set must be adjusted at every iteration.

$$\begin{aligned} &\min_{\boldsymbol{S}} \langle \nabla f(\boldsymbol{M}_l), \boldsymbol{S} \rangle \\ s.t. \quad &\boldsymbol{S} \in \mathbb{S}_d^+ \end{aligned} \tag{3.11}$$

The global linear minimization oracle is simple to optimize, as it consists of p Frobenius inner products between each $P(i,i) \quad \forall i \in \{1, .., d\}$ and the elements of the gradient. The linear minimization oracle for each local metric does not scale as easily, as there are p^2 atoms to consider. In cases where p is rather large, we use the heuristic proposed in [52]. The general idea of the heuristic is to randomly select a row of the matrix, and consider all columns. Once the column containing the lowest LMO score is found, that column is used while all rows are considered. This technique has order $O(p)$ complexity and we find that it works relatively well in practice.

Once an atom $\boldsymbol{S}_{FW}$ has been selected to increase the weight on, an atom must be selected to remove weight from (the "away" atom). This problem is extremely similar to the linear minimization oracle problem, and consists of calculating the inner Frobenius norm between the gradient and each atom in the active set, taking that which has the maximum value. This is shown as the following, with $\boldsymbol{G}$ representing the gradient at the current iterate.

$$\max_{\boldsymbol{S}} \langle \boldsymbol{G}, \boldsymbol{S} \rangle, \quad \forall \boldsymbol{S} \in \mathcal{A}$$

Within a small number of iterations of the algorithm, it is common for the number of active constraints to quickly drop to a fraction of the original number. Inactive constraints have no contribution to the gradient and may be ignored. We leverage this fact in our implementation which results in significant speedups.

We provide a Matlab implementation of our code available for download[1].

[1] `https://github.com/jstamand/`

3.4 Experiments

The proposed metric learning technique, Sparse Compositional Local Metric (SCLM) learning is evaluated under a classification scenario using a K-Nearest Neighbors(K-NN) approach. There are few sparse metric learning techniques capable of handling high dimensional data. The high dimensional similarity learning (HDSL) algorithm developed in [52] is selected as the current state of the art algorithm for comparison. The support vector machine (SVM) with a linear kernel is included in the experimental evaluations to serve as a baseline algorithm. In our experiments the implementation provided by the LIBSVM library [14] is used.

3.4.1 Datasets for Evaluation

Five datasets were selected to run the experimental evaluations on, the characteristics of these datasets are summarized in Table 3.1.

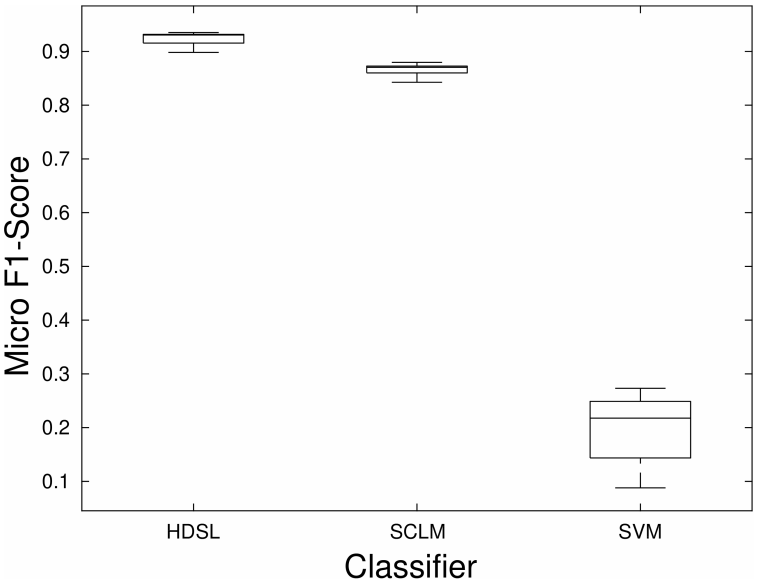

Figure 3.2: Reported micro-averaged F1 scores of classification experiment on CNAE-9 dataset.

3.4.1.1 CNAE-9

The classification nacional de atividae econamicas(CNAE) dataset[2] consists of 1080 documents of text business descriptions from Brazilian companies. Each of the companies is placed into one of nine categories, based on their national economic activities. The data is preprocessed such that punctuation and the most frequently occurring words are removed. Each document is then represented as a vector with each entry weighted occurring to word frequencies.

3.4.1.2 BBC-Sports & BBC-News

The BBC-Sports and BBC-News datasets[28][3] originate from BBC News. The BBC-Sports dataset consists of 737 documents, and the BBC-News dataset consists of 2225 documents. We use the provided pre-processed form, where each dataset is given as a term-document frequency matrix. Each of the two datasets have five distinct classes.

3.4.1.3 TDT2-30

The TDT2-30 dataset[4] is a subset of the original NIST Topic Detection and Tracking corpus. The version we use was prepared by [12], and contains only the 30 most frequently appearing labels. The dataset consists of 9,394 documents, each with 36,771 features.

3.4.1.4 Madelon

The Madelon[5] dataset is an artificial dataset created for the NIPS 2003 feature selection challenge. It consists of 2500 samples which form 32 separate clusters, each cluster is located on a vertice of a five-dimensional hypercube. The data points are randomly assigned a binary label, making this an especially challenging classification task.

[2] `https://archive.ics.uci.edu/ml/datasets/CNAE-9`
[3] `http://mlg.ucd.ie/datasets/bbc.html`
[4] `http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html`
[5] `https://archive.ics.uci.edu/ml/datasets/Madelon`

3.4.2 Classification Experimental Setup

A cross validation procedure is utilized to separate the model selection and model evaluation processes. The dataset is partitioned into training and testing sets using a five fold cross validation procedure. The training set is then further partitioned into additional training and validation sets using an internal round of cross validation. The training and validation sets are used optimize the model using a grid search over the parameter space. After the best performing model is located, it is trained on the combined training and validation sets, and predictions are made and evaluated on the test set. This is repeated for each of the five folds of cross-validated data.

The Support Vector Machine is trained using a linear kernel with the regularization parameter c is tuned to the best value such that $c \in \{1c-6, 1e-5, 1e-5, 1e-4, 1e-3, 1e-2, 1e-2, 1e0, 1e1, 1e2, 1e3, 1e4, 1e5\}$. The high dimensional similarity learning algorithm does not have many tunable parameters, we vary the scaling parameter γ such that $\gamma \in \{1e0, 1e1, 1e2, 1e3, 1e4\}$. The proposed algorithm has two tunable parameters $(\lambda_{global}, \lambda_{local})$ controlling the balance between local adaptivity and global consistency, we

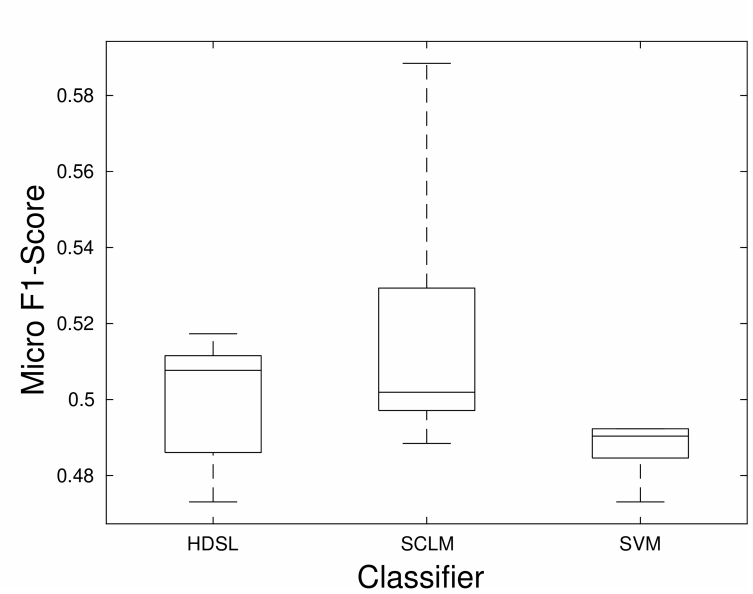

Figure 3.3: Reported micro-averaged F1 scores of classification experiment on Madelon dataset.

allow both parameters to vary on the set of $\{1e0, 1e1, 1e2, 1e3, 1e4\}$.

The proposed and state of the art techniques (SCLM and HDSL, respectively) both utilize a Frank-Wolfe style optimization technique. It is well known that Frank-Wolfe optimization converges at a sublinear rate when the solution lies on the boundary region [46]. We observed that for SCLM and HDSL, the objective value is quickly reduced in the first hundred or so iterations and then makes minimal progress towards the solutions. The maximum allowable number of iterations was set to 500 to ensure both algorithms terminated in a reasonable amount of time.

The SCLM algorithm relies on an external technique to provide the subset of samples which correspond to each local metric. In the experiments we cluster the data using unsupervised clustering methods. The CNAE-9, BBC-Sports, and Madelon datasets were clustered using the Sparse and Robust K-means Clustering (RSKC) algorithm of [44]. We found that RSKC had difficulty operating on the larger datasets, so the k-NN clustering algorithm with the cosine similarity function was utilized to cluster the BBC-News and TDT2-30 datasets.

The HDSL and SCLM algorithm share several similarities, our aim was to train them in the same fashion. Both algorithms are trained using relative distance constraints in the form of triplets, which are formed for each point using 3 friends (same-label points) and 8 impostors (different label points). After training, test points are classified using a k-NN approach based on the learned metric with $k = 3$.

We evaluate each classfier using the micro-averaged F1-score. The vanilla F1-score is the harmonic mean of precision and recall. Precision and recall may be calculated on a per-class basis by taking the predicted and true label values and calculating the number of positive (P) and negative (N) samples, and the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. One found, precision and recall are calculated as follows:

$$\text{precision} = \frac{TP}{TP + FP} \qquad \text{recall} = \frac{TP}{TP + FN}$$

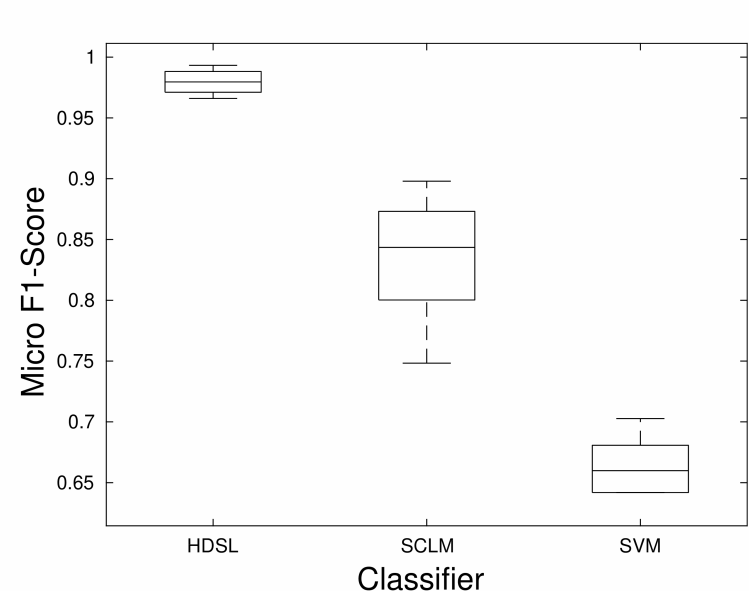

Figure 3.4: Reported micro-averaged F1 scores of classification experiment on BBC-Sports dataset.

The F1-score is the harmonic mean of precision and recall, and is calculated as shown below:

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.12}$$

Typically, the F1-score is calculated on a per-class basis, simple averaging may bias the score towards the classes which contain fewer samples. This effect is alleviated by calculating the micro-averaged F1-score, which is the F1-score found by calculating precision and recall in a global manner (by summing the TN, TP, FN, FP from each class). Finally, the micro-averaged F1-scores for each cross validation fold are averaged, which is how the reported metric is calculated for each experiment.

3.4.3 Classification Results

In the results, we refer to our proposed method as Sparse Compositional Local Metrics (SCLM) and the method of [52] as High Dimensional Sparse Learning (HDSL). When compared with the support vector machine baseline algorithm, our proposed technique was able to outperform it on all datasets except for BBC-News. Though support vector machines

typically perform well on sparse datasets, we found that it had a particularly hard time on the CNAE-9, Madelon and TDT2-30 datasets. It was observed that in these cases, it had a tendency to make predictions nearly all of one type, which brought the performance measure averages down significantly.

An inspection of the results show that for the CNAE-9, BBC-Sports, and TDT2-30 datasets, the HDSL method has a clear advantage. The Madelon dataset was particularly challenging, with HDSL and SCLM producing around the same level of performance on average. We note that the performance of SCLM on the Madelon dataset was extremely good for some folds, and comparative to the HDSL method on others. Finally, in the BBC-News dataset, the SCLM method appears to have the advantage. The complete performance results of all methods and datasets are shown as Figures 3.2, 3.3, 3.4, 3.5, and 3.6.

3.4.4 Visualization Experiment

To demonstrate the ability of the proposed distance metric to learn local trends in the data, we ran an experiment on the BBC-News dataset using three local metrics. After learning

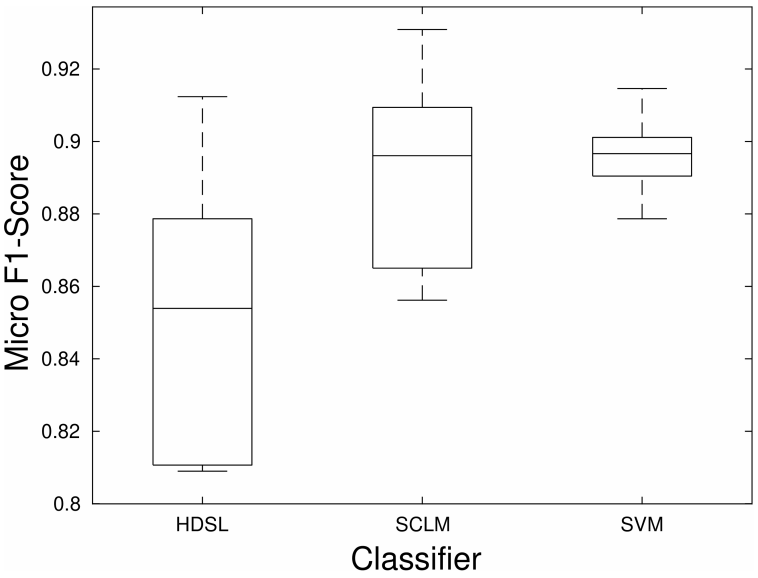

Figure 3.5: Reported micro-averaged F1 scores of classification experiment on BBC-News dataset.

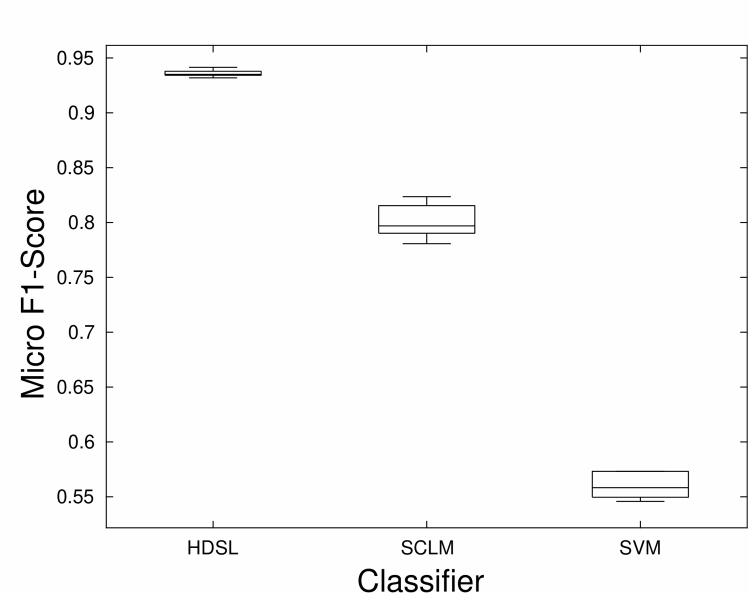

Figure 3.6: Reported micro-averaged F1 scores of classification experiment on TDT2-30 dataset.

the metrics, we extracted the (sparse) interactions from each local metric and created a "word-cloud" where the size of word reflects the influence of that word on the metric. This is shown as figure 3.7. We observe trends in the metrics of 3.7, the local metric #1 appears to honed in on "e-sports/video games", local metric #2 is focused more on digital entertainent (e.g. hiphop, music charts, cinema, portable playerm, etc), local metric #3 appears to be more political and budget oriented. The global metric does not appear to have any strong trends, which indicates that the local trends may not have much in common.

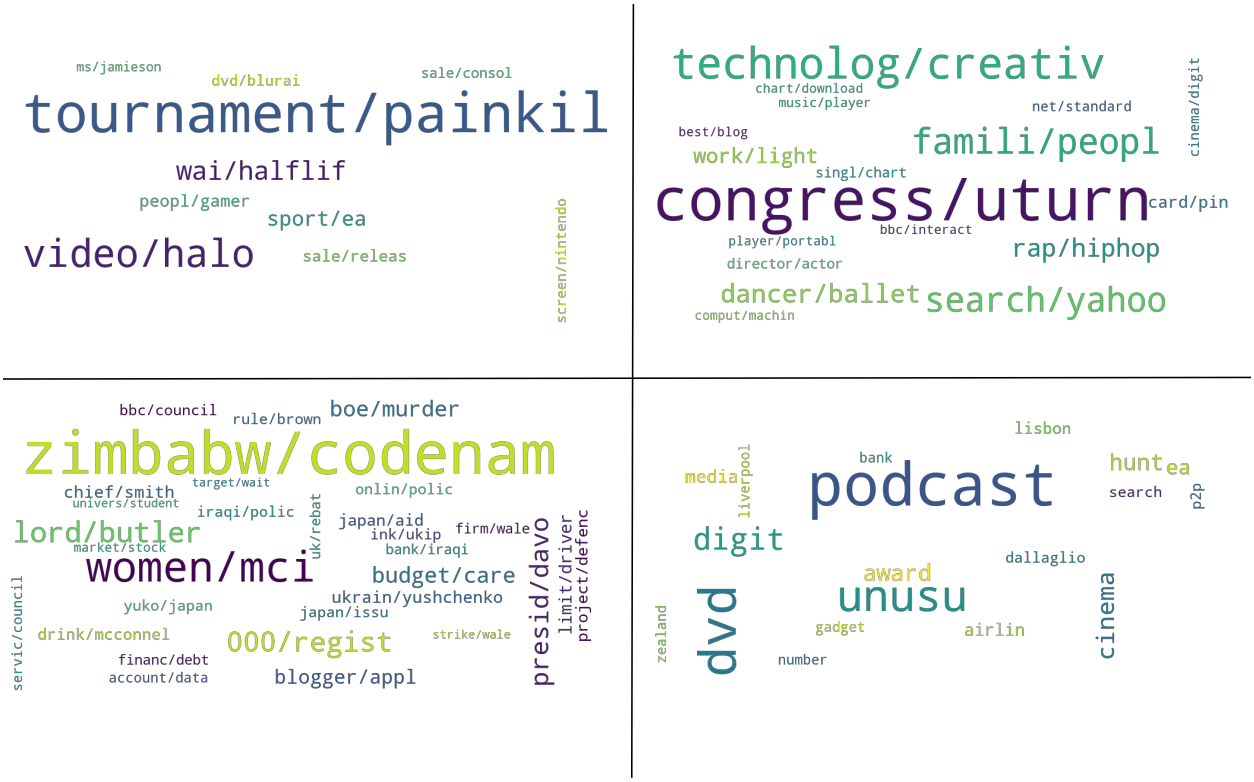

Figure 3.7: Word clouds for each of the local metrics learned on the BBC-News dataset. Top-Left: Local Metric #1, Top-Right: Local Metric #2, Bottom-Left: Local Metric #3, Bottom-Right: Global Metric

3.5 Discussion and Future Work

3.5.1 Impact of Top-level Clusterings

The HDSL method outperformed the SCLM method by a good margin on the TST2-30 and BBC-Sports datasets, in these cases the SCLM method demonstrated a larger variance. One source of this phenomenon could be the overlying clustering algorithm which determines the "local groupings" or clustering of the data. Clustering algorithms are typically not convex and are often sensitive to the initialization. Widely varying clusterings of the data between cross validation folds could be one source of the variance in classification performance.

In cases where SCML underperformed HDSL, one explanation could be that the data may not be multi-model. Given a dataset which exhibits a single mode, one can expect that splitting the data and training two (or more) classifiers will not produce the same results as training a single classifier on the complete data, especially in the case of a high-dimensional feature space. A less extreme version of this scenario could be a mismatch between the number of modes and the target number of local groups. In summary, the top-level clustering method has no knowledge of the labels and the metric to be learned, and could be creating local groups which are not useful.

A promising direction for future work is the integration of the top-level clustering procedure with the local metric learning algorithm. The ability to exchange information between the clustering and the local metrics could result in local groups which are meaningful with respect to the metric learning task.

3.5.2 Algorithm Acceleration

To accelerate the runtime of the HDSL algorithm and our own, we both make use of a stochastic estimate of the gradient. As noted in [52], the accuracy of this approximation is bounded and works relatively well in practice. However, we point out that this approximation also serves to prevent the gradient matrix from "filling-in". This is particularly noteworthy,

as it may be infeasible to store a non-sparse gradient matrix. For the diagonal matrix in our technique, we only calculate the diagonal elements of the gradient, upper bounding the space complexity to $O(p)$. However, for the local metric matrices(and that in [52]), as the number of constraints grows, it could be possible to get a full gradient matrix. Placing a limit on the number of constraints used in estimating the gradient could prevent this, depending on the sparsity level of each sample.

The proposed work and others [36, 52] make use of updates using "Jaggi atoms". Another promising direction of research could be the development of a specialized Frank-Wolfe technique which takes advantage of updates with this structure. This type of work would be widely applicable, as there are many positive semi-definite programming problems in machine learning.

3.6 Conclusion

In this chapter, we presented an algorithm for sparse compositional local metric learning, where each local metric consists of a local and shared global component. A pairwise alternating Frank-Wolfe style optimization algorithm was used to optimize the objective in an efficient projection-free manner. The proposed method was able to maintain sparsity of the solution through the optimization process, which allowed scaling to datasets with over 30,000 features. An empirical evaluation of the proposed technique was executed against a solid baseline algorithm and the current state of the art in sparse similarity metric learning. The results of the empirical evaluation demonstrate that our method is more effective than the baseline measure, and is comparable to the competing method for some datasets. Finally, a discussion was presented and directions for future work in this area was outlined.

Chapter 4

Robust Regressive Virtual Metric Learning with Structured Sparsity

4.1 Introduction

Distance metric learning is concerned with learning how objects are similar in a pairwise manner. The metric is typically in the form of the Mahalanobis distance function and is parameterized by a positive semidefinite matrix $\boldsymbol{M}$. The classic approach to distance metric learning suffers from two major computational restrictions; the vast number of pairwise constraints, and ensuring that $\boldsymbol{M}$ is positive semi-definite. Regressive Virtual Metric Learning (RVML) [59] is a new approach to distance metric learning based on regression. The advantage of RVML is that it only requires a linear number of constraints and $\boldsymbol{M}$ is learned in a projection-free manner. However, RVML has no mechanism to discard or reduce the influence of noisy, corrupted, or irrelevant features. In this chapter, we present a new method for robust regressive virtual distance metric learning with structured sparsity.

Under the classic metric learning approach, it is often desirable that the metric $\boldsymbol{M}$ be *low-rank*. A low-rank $\boldsymbol{M}$ limits the model complexity and in many cases improves the prediction accuracy. This is often referred to as *output sparsity*, as a low-rank $\boldsymbol{M}$ is equivalent to projecting the data into a low-dimensional subspace and measuring using the Euclidean distance, which typically results in a sparse representation in the output space. Previous works have focused on the concept of output sparsity [87]. However, the problem of optimizing a matrix subject to a low-rank constraint is a problem known to be NP-hard [5].

The regressive virtual metric learning approach has a natural mechanism for output sparsity, as the rank of $\boldsymbol{M}$ is upper-bounded by the trailing dimension of the virtual point matrix. If the dimension of the virtual points is low, this results in reduced model complexity without a low-rank constraint. Ideally, a low-rank $\boldsymbol{M}$ should induce a subspace where irrelevant features have little to no influence. It is known that popular metric learning algorithms often suffer rather severe performance degradation when non-informative or noisy features are introduced [50].

An alternative to output sparsity is that of *input sparsity*, where only a portion of the input features are selected by the metric. Because $\boldsymbol{M}$ must be positive semi-definite, most

metric learning algorithms take an iterative approach which alternates between updating $\boldsymbol{M}$ and then projecting $\boldsymbol{M}$ onto the positive semi-definite cone. Because a projection could destroy any input sparsity of $\boldsymbol{M}$ produced in the update, promoting sparsity of the input requires careful structuring of $\boldsymbol{M}$ such that the positive semi-definiteness is preserved during the update cycle. Several recent works accomplish this using a variety of structured update schemes ([36], [52], [69]). However, these methods are often slow and have been shown to converge at a sublinear rate [37].

In this chapter, we build upon the regressive virtual metric learning approach and propose a new regression-based distance metric learning method which is robust to the influence of irrelevant features. In our approach we can control the amount of input sparsity applied through our use of a penalty which induces structured sparsity of the metric. Our method is based on linear multiple regression and only requires constraints which are linear in the number of samples. Additionally, we can guarantee that our metric matrix $\boldsymbol{M}$ is positive semidefinite by construction. We present an empirical evaluation where we demonstrate the performance of our model, and show that the metric is often much more interpretable than other popular approaches. Specifically, in this chapter we claim the following contributions:

1. A new algorithm for robust regressive virtual metric learning with structured sparsity. Our proposed model learns a metric in an indirect manner and through use structured sparsity inducing penalties has the ability to select and reweight features.

2. We provide an experimental evaluation of our proposed approach on five noise-augmented real-world datasets in a classification scenario. This comparison demonstrates that even in the case of low-noise, our method is comparable to the current state of the art in terms of classification performance.

3. We provide a second empirical evaluation where we augment the datasets with irrelevant features. In this evaluation we demonstrate that in this scenario our method excels and is superior to the state of the art on all evaluation datasets.

4. Finally, we produce insightful visualizations to show each algorithms learned metric. This demonstrates that in most cases the proposed method is easy to interpret.

4.2 Background

Distance metric learning is a rich and varied topic that has been well-studied over the years. For a comprehensive overview we recommend the surveys [5, 45]. By far the most popular metric to learn is that of the Mahalanobis distance (equation. 4.1), where $\boldsymbol{M}$ defines the parameters of the metric and is positive semi-definite.

$$d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x} - \boldsymbol{y}) \tag{4.1}$$

The typical approach is to constrain the problem in a weakly-supervised manner by specifying points which are "must-link" and "cannot-link" in a pairwise manner, as shown in equation 4.2.

$$\begin{aligned} \mathcal{S} &= \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_i \text{ similar to } \boldsymbol{x}_j\} \\ \mathcal{D} &= \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_i \text{ similar to } \boldsymbol{x}_j\} \end{aligned} \tag{4.2}$$

4.2.1 Large-Margin Approach

One of the most popular and widely used approaches to metric learning is the Large Margin Nearest Neighbors (LMNN) method of [77]. LMNN extends pairwise constraints such that a third point is introduced. Given a triplet of samples $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ such that $\boldsymbol{x}_i, \boldsymbol{x}_j$ are similar and $\boldsymbol{x}_i, \boldsymbol{x}_k$ are dissimilar. The goal of the large margin approach is to ensure that the distance $d_{\boldsymbol{M}}(\cdot, \cdot)$ between the similar points is smaller than the distance between the dissimilar points by at least some margin amount m, as can be seen in equation 4.3.

$$\{d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) < d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_k) + m : (\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ similar}, (\boldsymbol{x}_i, \boldsymbol{x}_k) \text{ dissimilar}\} \tag{4.3}$$

A shared weakness of learning with pairwise and triplet constraints is the excessive number of constraints required. For example, in a two class classification scenario with n_A samples having label "A" and n_B samples having label "B", the total number of triplet constraints are $(n_A + n_B) \cdot n_A \cdot n_B$. Virtually all large-margin based methods approximate the constraints by only enforcing a select subset, as enforcing all of them is infeasible for even a moderate amount of samples.

Selecting a constraint subset is typically accomplished by limiting the number of constraints per points. The Euclidean distance is used to find the nearest neighbors for each point and form constraints only for those neighbors. It is well known that this approach to metric learning is particularly sensitive to the ability of the Euclidean distance to select good target neighbors to form constraints with.

4.2.2 Regression-Based Approach

A recent work by [59] proposes a new approach to distance metric learning within a regression-based framework. In lieu of forming pairwise or triplet distance constraints, the idea is to pull each instance to a separate "virtual point". The method takes an indirect approach to learning the metric and finds an $\boldsymbol{L}$ such that $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$, which may be solved for by minimizing a simple regression problem (equation 4.4). The solution of which has a convenient closed-form solution.‘

$$\min_{\boldsymbol{L}} f(\boldsymbol{L}, \boldsymbol{X}, \boldsymbol{V}) = \min_{\boldsymbol{L}} \frac{1}{n} ||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \lambda ||\boldsymbol{L}||_F^2 \tag{4.4}$$

Similar to pairwise or triplet constraints, the virtual points may be viewed as a form of weak supervision, as learning the metric does not interact with the labels directly. While the regression-based approach to learning the metric is rather straightforward, there is still the question of how to determine the virtual points.

In [59], two methods for determining the virtual points are proposed. The first being a

class-based representation, where the virtual point assigned to each instance is the one-hot encoding of that instances label. The second suggested method is to first extract a set of landmark points via a variation of the landmark selection problem [41] and then compute the optimal transport from the set of instances to the landmark points using the Sinkhorn-Knopp algorithm [17]. Unfortunately, the landmark selection heuristic depends on the ability of the Euclidean distance to select good landmarks. As the Euclidean distance treats all features equally, we believe the ability to select good landmarks may degrade as a function of how many features in the data are irrelevant.

A favorable characteristic of the regression-based approach is that the number of constraints are linear in the number of samples. Additionally, the metric parameters are learned indirectly through $\boldsymbol{L}$, where $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$. Meaning that $\boldsymbol{M}$ is positive semi-definite by construction. We identify some new opportunities for research in this area: 1) new forms of the metric $\boldsymbol{M}$; 2) other means for determining the virtual points. The work of this chapter is focused on the first point.

4.3 Related Work

Regressive Virtual Metric Learning (RVML) [59] is the only other regression-based approach to metric learning. The key difference between RVML and the proposed method is that RVML has no mechanism for discarding irrelevant features. It does has the potential to limit the rank of $\boldsymbol{M}$, but this ability is tied to the dimension of the virtual points and it is not clear how to manipulate that dimension. Additionally, low-rank constrained metrics do not typically excel in these situations.

The work of [86] proposed a Feature-AwaRe distance Metric (FARM), the metric consists of two parts $\boldsymbol{D}\boldsymbol{M}$ where $\boldsymbol{D}$ is a diagonal matrix. The formula in non-convex and requires an optimization strategy which updates $\boldsymbol{M}$ and $\boldsymbol{D}$ in an alternating fashion. Because of the non-convexity of FARM, it is particularly sensitive to the initialization of $\boldsymbol{D}$ and $\boldsymbol{M}$. The FARM method fits under the classic relative-distance metric learning approach and requires

constraints which are quadratic in the number of samples. Additionally, FARM learns $\boldsymbol{M}$ directly and must make a projection onto the positive semi-definite cone at every iteration.

4.4 Methodology

Our method operates under the typical multi-label learning scenario. Given a set of samples $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, the associated labels $\boldsymbol{Y} \in \mathbb{R}^{n}$ and virtual points $\boldsymbol{V} \in \mathbb{R}^{n \times q}$ we are interested in learning a regressive virtual type metric which is capable of feature selection. This requires that the metric matrix $\boldsymbol{M}$ is a combination of 4-sparse hermetian positive semi-definite matrices. This same structured sparsity pattern can be reproduced in terms of the Cholesky decomposition $\boldsymbol{L}$ of $\boldsymbol{M}$, where $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{T}$, by placing a group-sparse penalty[89] along each row of $\boldsymbol{L}$. This is shown as equation 4.5, where $\boldsymbol{L}(i,:)$ represents the ith row of $\boldsymbol{L}$.

$$\frac{1}{n}||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \lambda \sum_{i=1}^{p} ||\boldsymbol{L}(i,:)||_2 \tag{4.5}$$

To obtain the virtual points $\boldsymbol{V}$, we refer to [59, Section 3.3.1], where $\boldsymbol{V}$ is constructed as a class-based representation or learned by solving a problem of optimal transport. We call our proposed method Robust Regressive Virtual Metric Learning with Structured Sparsity (RRVMLSS) and present two variants, RRVMLSS-class and RRVMLSS-transport, which denote using virtual points found as a class-based representation or via solving a problem of optimal transport, respectively. Recall that the purpose of this work is to focus on the robustness of the metric $\boldsymbol{M}$ to irrelevant inputs, we leave other methods for determining $\boldsymbol{V}$ as a future research topic.

4.5 Algorithmic Robustness

The proposed method for metric learning may be interpreted as an instance of robust linear regression and as such is subject to guarantees in term of algorithmic robustness. The general problem of robust linear regression is to assume that the design matrix has been corrupted

by a possibly malicious source and then show that the method of interest is equivalent to a robust linear regression problem (equation 4.6).

$$\min_{\beta \in \mathbb{R}^m} \quad \max_{\boldsymbol{\Delta}} \quad ||\boldsymbol{y} - (\boldsymbol{A} + \boldsymbol{\Delta})\beta||_2 \tag{4.6}$$

We can show that equation 4.5 is equivalent to a problem of robust linear regression, all that is required is to rearrange the equation into a vector form. Let $y \in \mathbb{R}^{nq}$ and $\beta \in \mathbb{R}^{pq}$ be the vector equivalents of the matrices $\boldsymbol{A}$ and $\boldsymbol{L}$, respectively, that have been "flattened" by stacking the columns of each matrix. Let $\boldsymbol{A} \in \mathbb{R}^{nq \times pq}$ be a block-diagonal matrix equivalent of $\boldsymbol{X}$ where each block is a column of $\boldsymbol{L}$.

Let $\boldsymbol{\Delta}_g$ be a disturbance associated with group $g \in \mathcal{G}$, where $\mathcal{G}$ specifies the groupings as the rows of matrix $\boldsymbol{L}$. Also, let the norm of each group be upper bounded such that $||\boldsymbol{\Delta}_g|| \leq c_g, \quad \forall g \in \mathcal{G}$. Finally, define $\boldsymbol{\alpha} = [sign(\boldsymbol{\beta}_1) \cdot ||\Delta_1||_2, ..., sign(\boldsymbol{\beta}_{pq}) \cdot ||\Delta_{pq}||_2]$. With these assumptions consider the problem of robust linear regression (4.7) which is equivalent to equation 4.5 via a straightforward application of [83, Theorem 3].

$$\min_{\beta \in \mathbb{R}^{pq}} \left\{ ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2 + \max_{\forall g \in \mathcal{G}, ||\boldsymbol{\alpha}_g||_2 \leq c_g} \boldsymbol{\alpha}^T \boldsymbol{\beta} \right\} \tag{4.7}$$

4.5.1 Optimization

Equation 4.5 is convex with respect to $\boldsymbol{L}$ and may be minimized using a variety of off-the-shelf optimization techniques. We have selected a first order method called Fast Iterative Shrinkage-Thresholding Algorithm (FISTA). All that is required to apply FISTA is to calculate the gradient of the loss function and the proximal operator for the group-sparsity constraint. The gradient of the loss of 4.5, and the proximal operator for $\lambda \sum_{i=1}^{p} ||\boldsymbol{L}(i,:)||_2$ are both shown below as equations 4.8 and 4.9, respectively.

$$\frac{\partial f(\boldsymbol{L})}{\partial \boldsymbol{L}} = \frac{2}{n} \boldsymbol{X}^T (\boldsymbol{X} - \boldsymbol{V}) \tag{4.8}$$

$$p(\boldsymbol{L}) = \begin{cases} ||\boldsymbol{L}(i,:)||_2 \leq \lambda, \quad \boldsymbol{L}(i,:) = \mathbf{0} \\ \text{else}, \quad \boldsymbol{L}(i,:) = (1 - \frac{\lambda}{||\boldsymbol{L}(i,:)||_2}) \cdot \boldsymbol{L}(i,:) \end{cases} \tag{4.9}$$

We have implemented the FISTA optimization algorithm applied to equation 4.5 in C++, the implementation was bridged with Python using Cython[3] and a wrapper built for the algorithm to operator as a classifier in the Scikit-Learn [58] Python machine learning library. To foster research reproducability, we make an implementation of our algorithm available for download on github[1].

4.6 Experiments

4.6.1 Setup and Evaluation

In our experiments we compare against some baseline methods and also state of the regressive metric learning methods. The kNN algorithm with the Euclidean distance and $k = 1$ is used as our baseline metric. The Large-Margin Nearest-Neighbors (LMNN) [77] algorithm is arguably one of the most popular and effective metric learning algorithms and is chosen as a good representation of relative-distance based metric learning. In our experiments we configure the LMNN constraints by setting the number of similar samples to three and the number of dissimilar samples to five.

We also compare with RVML [59], which is the current state of the art in regression-based metric learning. We experiments with two variants of RVML known to perform well, RVML-class, and RVML-transport. These methods vary in how the virtual points are selected, RVML-class constructs the virtual points as a one-hot encoding of the labels. RVML-transport learns the virtual points as a problem of optimal transport. Both RVML variants contain a L_2 regularization parameter which we tune for best results in the range of 1×10^k, $k \in \{-5, -2, ..., 3\}$. The two variants of the proposed method, RRVMLSS-class and

[1] https://github.com/jstamand

Table 4.1: Summary of dataset statistics.

Dataset	**# Features**	**# Instances**	**# Classes**
Balance	4	625	3
Credit	15	690	2
German	20	1000	2
Segment	19	2310	7
Urban Land Cover	148	168	9

RRVMLSS-transport, each contain a regularization parameter controlling the level of group sparsity which we tune in the range of 1×10^k, $k \in \{-5, -2, ..., 3\}$.

In all experiments, we shuffle the dataset and partition into 60% training, 20% validation, and 20% test. Hyper-parameters are tuned for best performance using the training and validation partitions, for final evaluation each method is trained on the combined training and validation set, then evaluated on the test set. For each experiment this process is repeated 20 times and the average across the 20 trials is taken. We use a two-sided t-test to determine the significance of results. The performance metric we use for evaluation is the F1-score, as a good F1-score requires both good recall and precision. For datasets with more than one possible label we report the micro-averaged F1-score (micro-F1). This process is followed for both of the following experiments.

4.6.2 Classification Scenario

The purpose of our first experiment is to demonstrate the performance of the proposed method on datasets on which metric learning is known to perform well. We have selected five datasets from the UCI machine learning repository from a variety of domains; Balance, Credit, German, Segment, and Urban Land Cover. Summary statistics (e.g. # features) for these datasets are shown in Table 4.1.

Table 4.2: Summary of experimental results, displayed metric is the micro-F1 score averaged over 20 trials. The best results as measured by the F1-score (which is micro-averaged in the case of multi-class data) for each dataset are highlighted in **bold**. The number of stars(*) denotes the significance level as measured by the p-value in a two-sided t-test, $* \Rightarrow (\mathbf{p < 0.05})$ and $** \Rightarrow (\mathbf{p < 0.01})$.

	KNN	LMNN	RVML-class	RVML-transport	RRVMLSS-class	RRVMLSS-transport
Balance	0.9222	**0.9462****	0.9030	0.9158	0.9022	0.9182
Credit	0.6778	0.7377	0.6848	0.7179	0.7949	**0.8067****
German	0.6291	**0.6755***	0.6414	0.6518	0.5240	0.5711
Segment	0.7095	**0.8464**	0.8143	0.6762	0.8310	0.7281
Urban Land Cover	0.3348	0.6352	0.5007	0.4194	**0.7281****	0.6654

Table 4.3: Summary of experimental results on noise augmented data, displayed metric is the micro-F1 score averaged over 20 trials. The best results as measured by the F1-score (which is micro-averaged in the case of multi-class data) for each dataset are highlighted in **bold**. The number of stars(*) denotes the significance level as measured by the p-value in a two-sided t-test, $* \Rightarrow (\mathbf{p < 0.05})$ and $** \Rightarrow (\mathbf{p < 0.01})$.

	KNN	LMNN	RVML-class	RVML-transport	RRVMLSS-class	RRVMLSS-transport
Balance	0.8286	0.8848	0.8984	0.9016	0.9026	**0.9100***
Credit	0.6543	0.7364	0.6721	0.7100	0.7717	**0.7911****
German	0.6210	0.6806	0.6381	0.6415	0.6153	**0.6623****
Segment	0.7071	0.8500	0.8387	0.6744	**0.8804****	0.7423
Urban Land Cover	0.3365	0.6565	0.5172	0.4341	**0.7356****	0.6580

4.6.3 Classification Scenario with Noise Augmented Data

In the second experiment our aim was to demonstrate the robustness of our algorithm in the presence of noisy or irrelevant features. We have taken the datasets of Table 4.1 and augmented them with uninformative features. Each augment feature is generated by drawing from a normal distribution, with each distribution centered around a vertex of a p-dimensional hypercube. Each dataset was augmented with a number of features which is equal to the number of features it already contains, the result being that after augmentation each dataset contains twice as many features than it did previously.

After completing the classification portion of the experiment, for each dataset we took the matrix parameterizing each metric ($\boldsymbol{M}$) and plotted them to observe the features selected by the metric. In this visualization activity we expected that the robustness of each metric should relate to the propensity it has in selecting features in the upper-left quadrant, as selections in the other three quadrants indicate the selection of uninformative features.

4.6.4 Results and Discussion

The results of the classification scenario experiment (section 4.6.2) are shown in Table 4.2. The LMNN method provided the best performance on three out of the five datasets with two results being statistically significant. The proposed methods produced the best performance on two datasets, with significant results on both. We observe that on four out of the five datasets, the proposed methods (RRVMLSS-class and RRVMLSS-transport) produce approximately equal or better performance to their regression-based counterparts (RVML-class and RVML-transport). We attribute this to the ability of the proposed methods to better filter out noisy or irrelevant features occurring naturally in the data.

The results of the classification experiment on the noise-augmented data are shown in Table 4.3. We found that the proposed algorithms were able to outperform all of the competing methods. RRVMLSS-transport produced the best performance on three of the datasets, while RRVMLSS-class performed best on the remaining two. Four of the results were significant on the 1% level and the fifth result was significant on the 5% level.

Although one of the proposed methods always produces the best results on the noise-augmented data, we observed that the other proposed method sometimes produces poor performance. A good example is the experiment on the noise-augmented German dataset, where RRVMLSS-transport performed the best out of all methods, and RRVMLSS-class performed the worst. We observed similar trends for the previous works in regressive virtual metric learning (RVML-transport and RVML-class). This leads us to believe that the virtual points play a significant role in the performance of the model.

The plots visualizing the best performing metrics for each method on the Balance and Segment datasets are shown as Figures 4.1 and 4.2 respectively. We also produced figures for the remaining datasets, these may be found in the chapter Appendix. We observed that on the Balance and Segment datasets, the proposed methods are clearly the most interpretable and select the least amount of features. We observed that in most cases the proposed methods are able to produce sparser metrics than competing methods. The exceptions to

this are RRVMLSS-class on the German dataset (Figure 4.4), and RRVMLSS-transport on the Urban Cover dataset (Figure 4.5). The result of the RRVMLSS-transport method on the Urban Cover dataset is especially interesting as it has selected almost all features. We see that it has induced sparsity specifically in the upper-left quadrant and has selected all augment features. We interpret this as some select features in the data being more harmful for classification than the augment features. There are some other similar cases (see [29]) where methods have selected all features including irrelevant or generated ones, and we still able to produce good results.

4.7 Conclusion and Future Work

In this chapter, we proposed a new structured sparse metric learning technique with virtual regression. Our method builds upon the current state of-the-art in regressive virtual metric learning through the introduction of structured sparsity. This gives our method the ability to select informative features while discarding irrelevant or noisy ones. In our experimental evaluation, we demonstrated the efficacy of our method in a predictive task against the current state of-the-art. When presented with noise-augmented data, our method produced extremely good performance.

We identify several avenues for future work based on the scalability, virtual points, and structure of the metric. With the recent focus on big-data methods, we believe our regression based metric learning with structured sparsity could be a prime candidate for big-data applications. Further research could be done using graphics processing units(GPU) for acceleration, or with a distributed implementation under a framework like SPARK [90].

Other opportunities for further research include other methods for determining the virtual points, especially in the case of high dimensional feature spaces, where the concentration of distances effect comes in to play. Finally, traditional large-margin metric learning techniques have explored many different forms of structure in a distance metric [5]. It may be beneficial to explore these under the regression-based metric learning framework where structure may

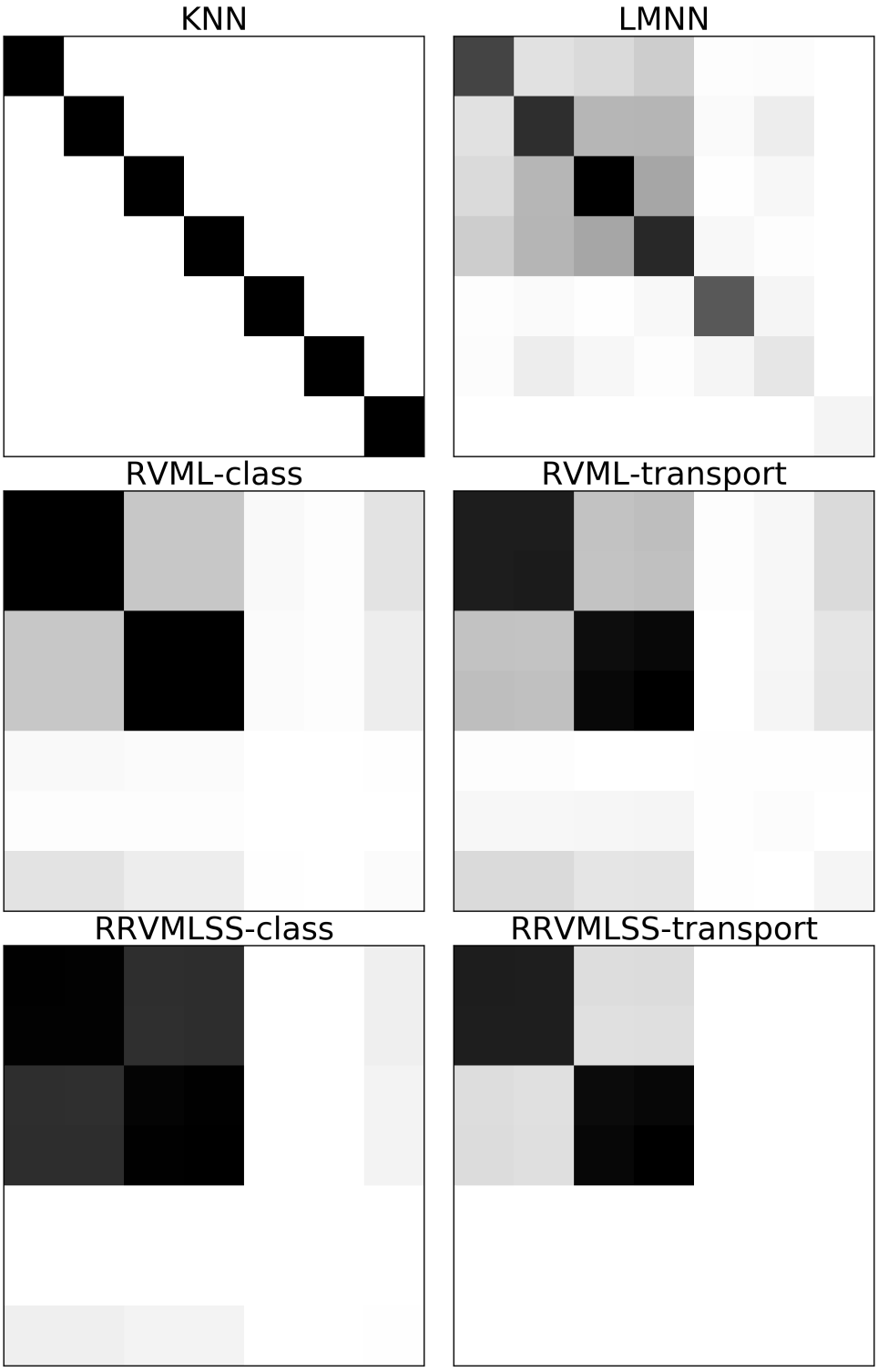

Figure 4.1: Graphical representation of metric learned on noise-augmented Balance dataset.

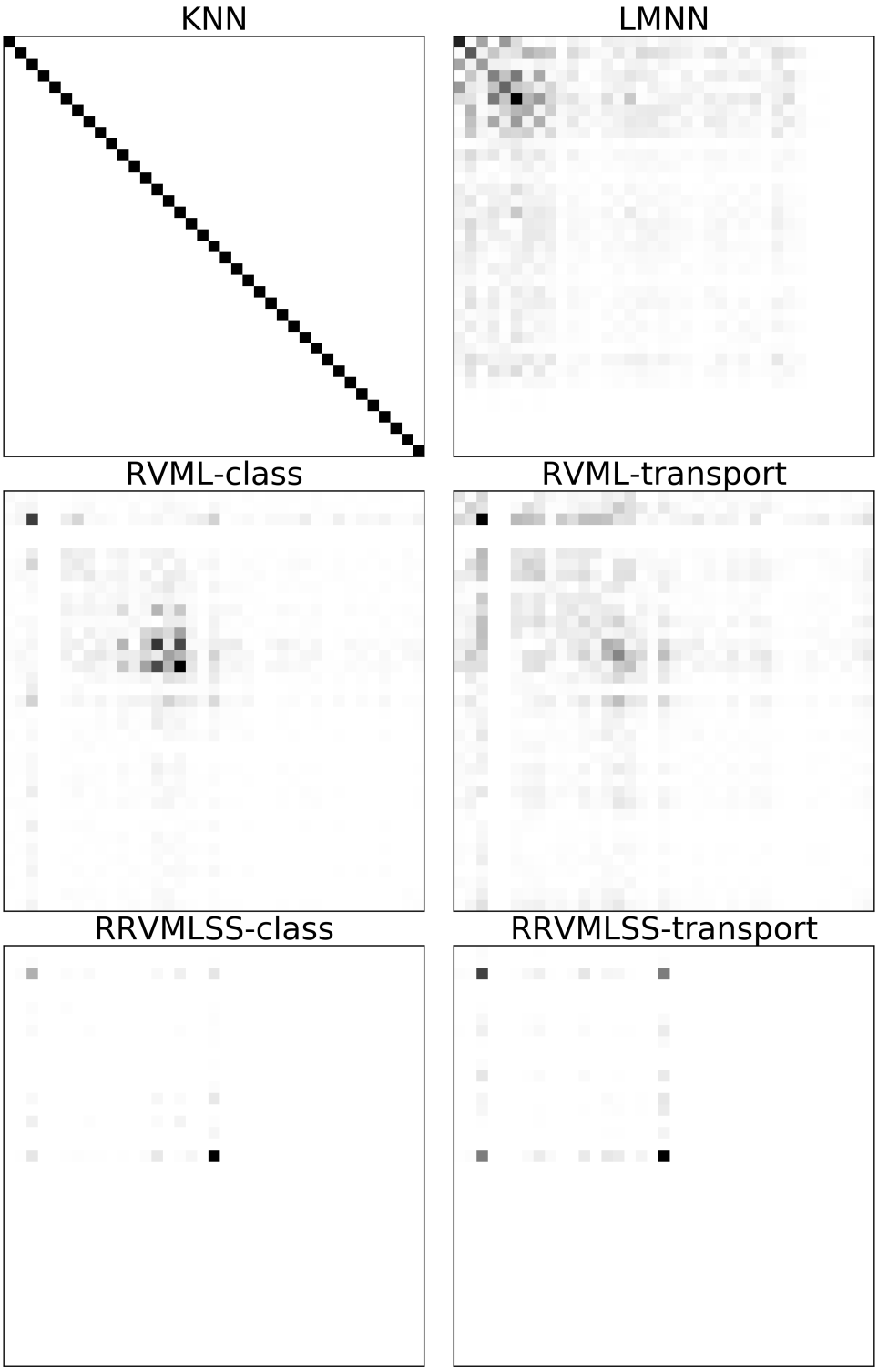

Figure 4.2: Graphical representation of metric learned on noise-augmented Segment dataset.

be present in both the virtual points and the metric.

4.8 Appendix

4.8.1 Metric Visualization

The appendix contains additional figures visualizing the learned matrices on the noise-augmented data experiments. Figures are shown for datasets Credit (Figure 4.3), German (Figure 4.4), and Urban Cover (Figure 4.5).

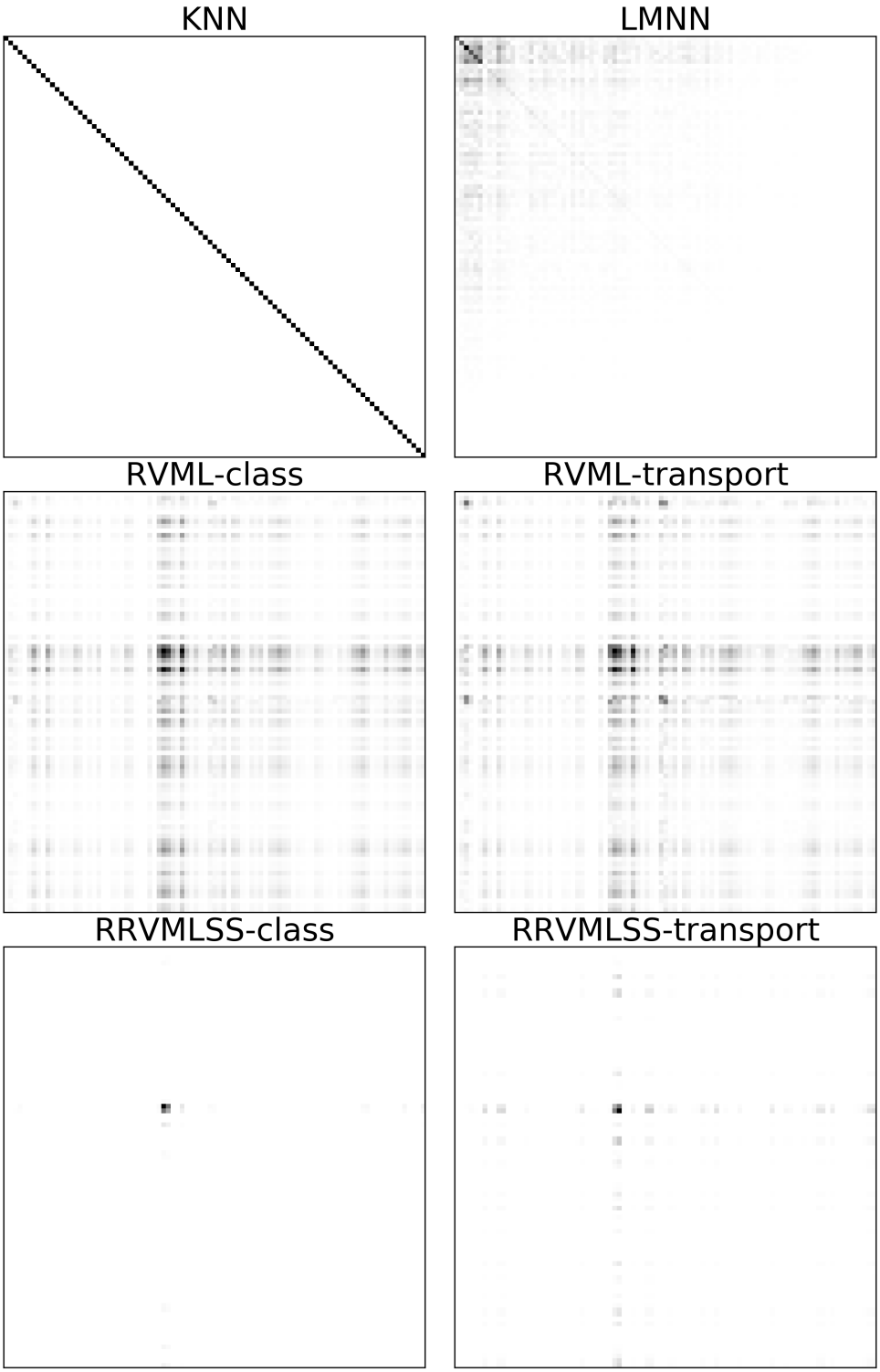

Figure 4.3: Graphical representation of metric learned on noise-augmented Credit dataset. Figure may be difficult to view due to large matrix size. LMNN, RVML-class, RVML-transport and RRVMLSS-transport all have entries in the upper left quadrant.

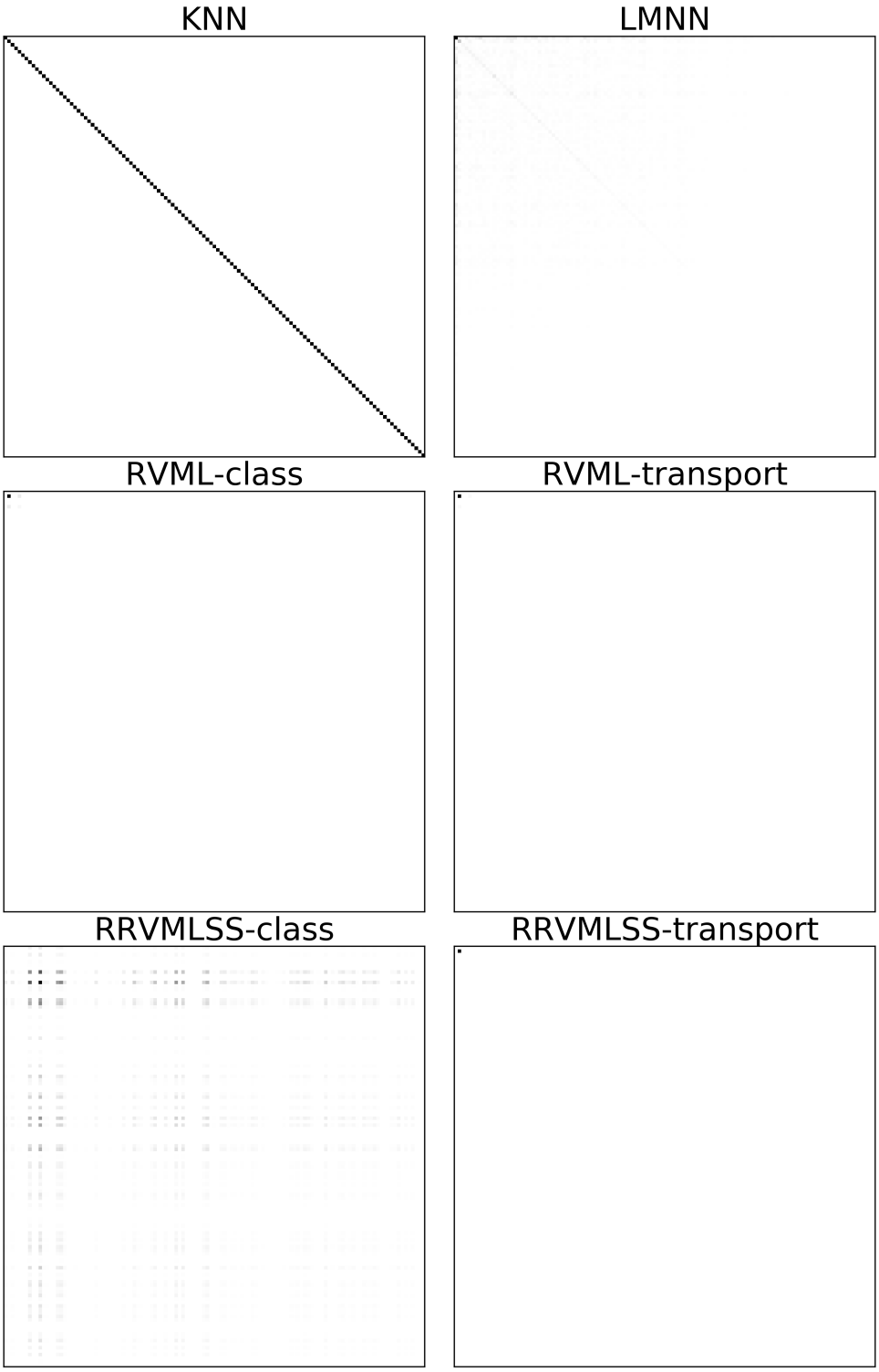

Figure 4.4: Graphical representation of metric learned on noise-augmented German dataset. Figure may be difficult to view due to large matrix size. LMNN, RVML-class, RVML-transport and RRVMLSS-transport all have entries in the upper left quadrant.

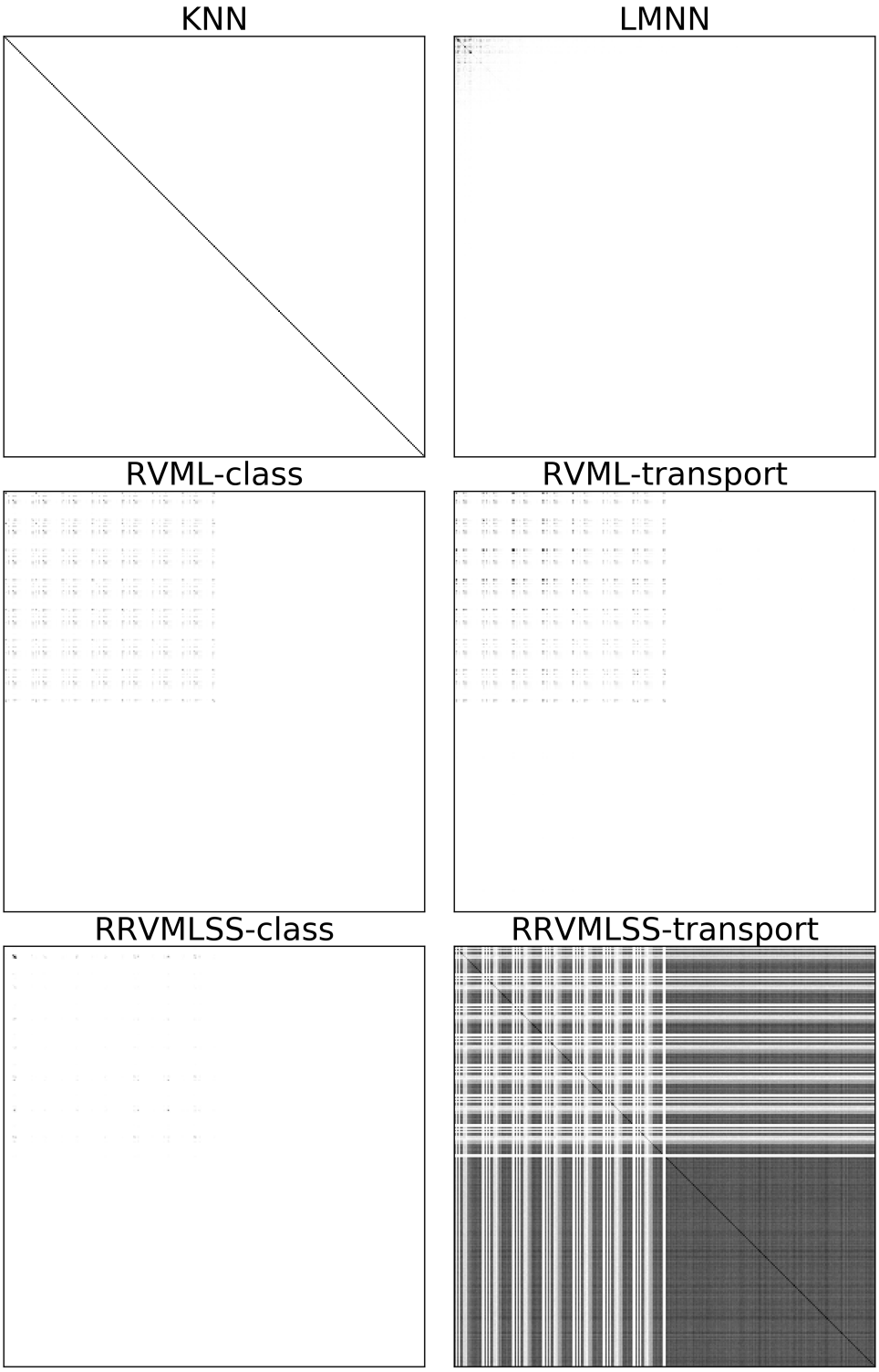

Figure 4.5: Graphical representation of metric learned on noise-augmented Urban Cover dataset.

Chapter 5

Regressive Virtual Regressive Metric Learning with Dynamic Margins

5.1 Introduction

Regressive virtual metric learning [59] is a new approach to learning a distance metric within a regression-based framework. The regression-based approach to distance metric learning requires constraints which are only linear in the number of instances. This gives it an advantage over the more traditional relative distance or "large-margin" based approaches (see [77] and derivative works). In lieu of enforcing relative distance constraints between instances, each instance under the learned metric should instead be located near a corresponding "virtual point". In this chapter, we propose a new approach to learning the virtual points which consider the factors of label discrimination and the compatibility of the virtual points with the metric.

The current state of the art in regressive virtual metric learning [59] employs three techniques to learn the virtual points: 1) virtual point selection via a class-based representation, 2) learning of the virtual points via a problem of Optimal Transport [17, 73], 3) random selection of virtual points[1]. Our main concern with the state of the art is that the methods to obtain the virtual points and learn the distance metric are treated as two separate processes. The weakness of this approach is that it admits no feedback mechanism from the distance metric, learning the virtual points is treated as an isolated process. A contribution of this chapter is the integration of the virtual point and distance metric learning processes into a single objective. This allows the distance metric to influence the virtual point locations during the learning process and vice-versa.

We make the following observation which serves to highlight the drawbacks of current methods in obtaining virtual points. The placement of the virtual points determines the *discriminative potential* of the distance metric, while the actual discriminative power is determined by the ability of the metric to transport[2] the instances to their associated points with low error (*compatibility*). Current approaches do not explicitly consider the compatibil-

[1]This technique is included in the supplementary material of [59]

[2]We often find it useful to talk about the metric in terms of the associated embedding ($\boldsymbol{L}$ in $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$), where $\boldsymbol{M}$ parameterizes the distance metric.

ity between the virtual points and the metric. In these cases the mapping from the instances to virtual points may be highly nonlinear and can never be satisfied by a linear metric. One may consider using the "kernel trick" to introduce a nonlinear metric, we point out that under the Akaike and Bayesian Information Criterion simpler models are often preferred [11]. As such, in this chapter we only consider the case of linear distance metrics, and leave more complex metrics for future work.

Learning the virtual points via optimal transport[59] is dependent on a variation of the landmark selection heuristic of [41] and relies on the ability of the Euclidean distance to select good landmarks. It is well known that classic metric learning methods have a particular sensitivity to the ability of the Euclidean distance to select good target neighbors [5]. One explanation for this is that the Euclidean distance has no mechanism to reject noisy or irrelevant features. We believe that the landmark selection heuristic has the same sensitivity to the Euclidean distance, which could render learning the virtual points via optimal transport as ineffective. In our proposed method, we obtain the virtual points as part of the process of learning the metric. As such, our method has no potential sensitivity to, nor relies on a naive metric like the Euclidean distance for initialization.

To address the limitations of regressive virtual metric learning, we propose a new innovative method which learns the virtual points and the metric jointly. In this method we explicitly integrate a label discrimination factor and a virtual point compatibility factor into the loss function of our distance metric. The discrimination factor takes into account the label value associated with each virtual point and encourages separation between points of different labels. The compatibility factor encourages the virtual points to be an approximately linear transformation of the instances, which is done by constructing the virtual points as a sparse representation of the input data. A notable innovation in this approach is that the responsibility of rejecting noisy or irrelevant features is taken up by the virtual points. To the best of our knowledge this contrasts all other works in metric learning, where noise rejection is accomplished by placing a regularization penalty on the metric, or

enforcement of the metric to be low-rank.

Specifically, in this chapter we claim the following contributions:

- A new method for regressive virtual metric learning which smoothly integrates the learning of the virtual points and the distance metric under a single multi-convex loss function.

- A theoretical analysis of our method demonstrating consistency where appropriate. We highlight the connection between our method an classical metric learning and show that our algorithm may be interpreted under the classical metric learning setting as a *dynamic margin adjustment*.

- A comprehensive empirical evaluation with the state of the art regressive virtual metric learning methods. We demonstrate the effectiveness of our algorithm in a classification scenario. We also demonstrate competitive scaling abilities, and an insightful visualizing under a clustering scenario.

- In the spirit of reproducible research, we provide a freely available implementation of our method for download[3].

[3]See section 5.5.1.

5.2 Background

Distance metric learning is a rich and varied topic that has been well-studied over the years, for a comprehensive overview we refer to the surveys [5, 45]. The main objective in metric learning is to learn a distance function $d(\cdot,\cdot) : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^+$ mapping two vectors to a distance in an effective manner. The effectiveness of a metric is determined by the ability of the metric to produce a small distance given similar examples (sharing same label) and a larger distance given dissimilar (having different labels) examples.

Much recent work [25, 26, 75, 76] has focused on the Mahalanobis [55] distance function (equation 5.1), where $\boldsymbol{M}$ must be positive semidefinite to ensure the properties of the distance metric. Learning $\boldsymbol{M}$ directly is computationally expensive due to the positive semidefiniteness of $\boldsymbol{M}$. Most techniques project $\boldsymbol{M}$ to the feasible constraint set with cubic cost, though there are some methods with clever update schemes which avoid this projection entirely [69]. A popular alternative is to factorize $\boldsymbol{M}$ such that the distance function is $(\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{L}\boldsymbol{L}^T(\boldsymbol{x}-\boldsymbol{y}) = ||\boldsymbol{x}^T\boldsymbol{L} - \boldsymbol{y}^T\boldsymbol{L}||^2$. An indirect approach to learning $\boldsymbol{M}$ is to learn $\boldsymbol{L}$, this avoids additional computation as $\boldsymbol{M}$ is then positive semidefinite by construction.

$$d_{\boldsymbol{M}}(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{y}) \tag{5.1}$$

One approach to learning the metric is to guide the learning process via pairwise constraints (equation 5.2), specifying which points are "must-link" and "cannot-link" via similarity/dissimilarity sets ($\mathcal{S}$ and $\mathcal{D}$ respectively).

$$\begin{aligned}\mathcal{S} &= \{(\boldsymbol{x}_i,\boldsymbol{x}_j) : \boldsymbol{x}_i \text{ similar to } \boldsymbol{x}_j\}\\ \mathcal{D} &= \{(\boldsymbol{x}_i,\boldsymbol{x}_j) : \boldsymbol{x}_i \text{ similar to } \boldsymbol{x}_j\}\end{aligned} \tag{5.2}$$

5.2.1 Large-Margin Approach

One of the most popular and widely used approaches to distance metric learning is the Large Margin Nearest Neighbors (LMNN) method [75, 76, 77]. LMNN extends pairwise constraints

such that a third point is introduced. Given a "triplet" of samples $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ such that $\boldsymbol{x}_i, \boldsymbol{x}_j$ are similar and $\boldsymbol{x}_i, \boldsymbol{x}_k$ are dissimilar. The goal of the large margin approach is to ensure that the distance $d_{\boldsymbol{M}}(\cdot, \cdot)$ between the similar points is smaller than the distance between the dissimilar points by at least some margin amount m, as can be seen in equation 5.3.

$$\begin{aligned} &\{d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) < d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_k) + m \\ &: (\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ similar}, (\boldsymbol{x}_i, \boldsymbol{x}_k) \text{ dissimilar}\} \end{aligned} \tag{5.3}$$

A shared weakness of learning with pairwise and triplet constraints is the excessive number of constraints required. For example, in a two class classification scenario with n_A samples having label "A" and n_B samples having label "B", the total number of triplet constraints are $(n_A + n_B) \cdot n_A \cdot n_B$. Virtually all large-margin based methods approximate the constraints by only enforcing a select subset, as enforcing all of them is infeasible for even a moderate amount of samples. This is done by retaining a set of close "target neighbors" for each point, as measured by the Euclidean distance. A well known vulnerability of LMNN is particularly sensitive to the ability of the Euclidean distance to select relevant target neighbors [5].

5.2.2 Regression-Based Approach

A recent work by [59] proposes a novel approach to distance metric learning within a regression-based framework. In lieu of forming pairwise or triplet distance constraints, the aim is to pull each instance to an associated "virtual point". The method takes an indirect approach to learning the metric and finds an $\boldsymbol{L}$ such that $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$. This indirect approach conveniently makes $\boldsymbol{M}$ positive semidefinite by construction. Solving for $\boldsymbol{L}$ is easily done by minimizing a simple regression problem (equation 5.4) which has a closed-form solution.

$$\min_{\boldsymbol{L}} f(\boldsymbol{L}, \boldsymbol{X}, \boldsymbol{V}) = \min_{\boldsymbol{L}} \frac{1}{n} ||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \lambda ||\boldsymbol{L}||_F^2 \tag{5.4}$$

Akin to pairwise or triplet constraints in the traditional approach, the virtual points may

be viewed as a form of weak supervision or side information. In [59], several methods for determining the virtual points are proposed. The first is a class-based representation, where the virtual points are a one-hot encoding of the labels. The second suggested method is to first extract a set of landmark points via a variation of the landmark selection heuristic [41] and then compute the optimal transport from the set of instances to the landmark points using the Sinkhorn-Knopp algorithm [17].

The connection between traditional and regression-based metric learning is highlighted by the following theorem from [59], showing that the risk of the traditional metric learning loss may be bounded by the empirical risk of the regressive approach (denoted by $\hat{R}(\boldsymbol{L})$) multiplied by a constant.

Theorem 5.2.1. *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{V} \subset \mathbb{R}^{d'}$ be a finite set of virtual points and $f_{\boldsymbol{v}}$ is defined as $f_{\boldsymbol{v}}(\boldsymbol{x}_i, y_i) = \boldsymbol{v}_i, \boldsymbol{v}_i \in \mathcal{V}$.*

Let $||\boldsymbol{v}||_2 \leq C_{\boldsymbol{v}}$ for any $\boldsymbol{v} \in \mathcal{V}$ and $||\boldsymbol{x}||_2 \leq C_{\boldsymbol{x}}$ for any $\boldsymbol{x} \in \mathcal{X}$. Let $\gamma_1 = 2max_{\boldsymbol{x}_k, \boldsymbol{x}_l, y_{kl}=1} d^2(\boldsymbol{v}_k, \boldsymbol{v}_l)$ and $\gamma_{-1} = 2max_{\boldsymbol{x}_k, \boldsymbol{x}_l, y_{kl}=1} d^2(\boldsymbol{v}_k, \boldsymbol{v}_l)$, we have that:

$$\begin{aligned} &\mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim \mathcal{D}, (\boldsymbol{x}_j, y_j) \sim \mathcal{D}} \left[y_{ij} (d^2(\boldsymbol{L}^T \boldsymbol{x}_i, \boldsymbol{L}^T \boldsymbol{x}_j)) - \gamma_{y_{ij}} \right]_+ \\ &\leq 8 \left(\hat{R}(\boldsymbol{L}) + \frac{8 C_{\boldsymbol{v}}^2 C_{\boldsymbol{x}}^2}{\lambda n} \left(1 + \frac{C_{\boldsymbol{x}}}{\sqrt{\lambda}} \right)^2 + \left(\frac{16 C_{\boldsymbol{x}}^2}{\lambda} + 1 \right) C_{\boldsymbol{v}}^2 \left(1 + \frac{C_{\boldsymbol{x}}}{\sqrt{\lambda}} \right)^2 \sqrt{\frac{ln \frac{1}{\delta}}{2n}} \right) \end{aligned} \tag{5.5}$$

A key insight is gleaned by noting that in theorem 5.2.1, the ideal margins in the classic formulation correspond to the distances between virtual points in the regressive formulation. Similar examples are located in a hypersphere of diameter $\hat{\gamma_1} = 2 \cdot max_{(\boldsymbol{x}, \boldsymbol{v})} ||\boldsymbol{x}^T \boldsymbol{L} - \boldsymbol{v}^T||_2$, and distances between hyperspheres of dissimilar examples is $\hat{\gamma_{-1}} = 2 \cdot min_{\boldsymbol{v}, \boldsymbol{v}', \boldsymbol{v} \neq \boldsymbol{v}'} ||\boldsymbol{x}^T \boldsymbol{L} - \boldsymbol{v}^T||_2$.

5.3 Related Work

Regressive virtual metric learning is a relatively new topic and to the best of our knowledge [59] is only closely related work. The primary innovations of [59] are the proposal to use virtual points to learn a distance metric in lieu of pairwise or triplet constraints, and the theory connecting the regressive and classical metric learning approaches. Three[4] techniques are suggested to obtain the virtual points; 1) selection via class-based representation, 2) learning virtual points via a problem of optimal transport, 3) random selection of virtual points. The key differentiating factor from our work is that none of these techniques to obtain the virtual points integrate feedback from the metric during the learning procedure.

We mention a related work in sparse coding [38]. In [38], the goal is to learn a discriminative sparse coding and a classifier in a simultaneous manner. Our work shares some similarities, namely the use of a label discrimination matrix and the joint learning of a sparse representation and classifier/regressor. However, there several significant differences. In [38], the sparse representation plays the role of the design matrix in the classification loss, the end goal is to learn a discriminative classifier under the strongly-supervised learning setting. In our work, the sparse representation plays the role of the labels in our regression loss. Our end goal is to learn a discriminative distance metric, we do not use the labels directly – our method operates under the weakly supervised learning setting.

[4]Two in main paper, one in the supplementary material.

5.4 Methodology

5.4.1 Overview

Given the design matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, each instance $\boldsymbol{x} \in \mathbb{R}^p$ needs to be mapped to a corresponding virtual point $\boldsymbol{v} \in \mathbb{R}^k$. The mapping is given as $\boldsymbol{L} \in \mathbb{R}^{p \times k}$ and corresponds to the metric matrix $\boldsymbol{M} \in \mathbb{R}^{p \times p}$ which is positive semi-definite by construction ($\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$). The primary goal of this work is to simultaneously learn the distance metric indirectly via $\boldsymbol{L}$ and the virtual points $\boldsymbol{V} \in \mathbb{R}^{n \times k}$.

In our method, we balance the potential discriminability of the metric by encouraging label consistency in the virtual points. Likewise, we wish for the metric to meet the "target level" of discriminability specified by the virtual points[5]. Our observation is that the instances need to be transported to their associated virtual points under the learned metric with relatively low error, otherwise they learned metric may not be very informative or useful.

5.4.2 Model

We begin with the regressive virtual metric learning loss of [59], shown as equation 5.6. The aim is to learn the distance metric through an indirect approach by optimization of $\boldsymbol{L}$.

$$||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 \tag{5.6}$$

Next, we consider the characteristics of the virtual points and make the following observation: under the well-known concentration of distances phenomenon [7] the virtual points lose their discriminative power as the dimension of each $\boldsymbol{v} \in \boldsymbol{V}$ increases. We place a sparsity inducing regularization penalty on each virtual point. This shifts the responsibility of rejecting noisy and irrelevant features to the virtual points. To the best of our knowledge this is a paradigm shift from previous works in metric learning, where the responsibility of

[5] We refer to this as *compatibility*.

noise reduction is taken up by regularization of $\boldsymbol{L}$ or the application of the trace norm to make $\boldsymbol{L}$ low-rank [34].

Considering the compatibility aspect, we may expect an approximately linear relationship from $\boldsymbol{X} \rightarrow \boldsymbol{V}$ (and vice-versa). We consider a reverse view of equation 5.6 where the objective is to predict the data using the virtual points by minimization of $||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2$. Combining this term with equation 5.6, and the proposed regularization on $\boldsymbol{V}$, yields equation 5.7.

$$||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \sum_{\boldsymbol{v} \in \boldsymbol{V}} ||\boldsymbol{v}||_1 + ||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2 \tag{5.7}$$

Inspecting equation 5.7, one may observe that the terms $||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2 + \sum_{\boldsymbol{v} \in \boldsymbol{V}} ||\boldsymbol{v}||_1$ are an approximation of the well-known problem of dictionary learning and sparse coding [2]. We take inspiration from recent work in the sparse coding literature [38] and introduce a final term to our model to promote label discrimination in the virtual points (equation 5.8).

$$\begin{aligned} \underset{\boldsymbol{L},\boldsymbol{V},\boldsymbol{A}}{\text{minimize}} \quad & ||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + ||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2 + ||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2 \\ s.t. \quad & ||\boldsymbol{v}_i||_1 \leq T_{\boldsymbol{V}} \quad \forall i \end{aligned} \tag{5.8}$$

In equation 5.8, the matrix $\boldsymbol{Q}$ promotes the virtual points to be discriminative with respect to the labels[6]. In the term $||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2$, $\boldsymbol{Q}$ is best viewed as the "ideal" discriminative virtual points, $\boldsymbol{A}$ introduces some flexibility into the model by allowing the virtual points to be only a linear transformation away from ideally discriminative. This added flexibility may allow the model to assign different amounts of separation between points with different labels. For example, in an image classification scenario, we may expect a larger separation between "fish" and "bird" images than between "fish" and "whale" images.

To summarize the model, we present a concise description of each term below[7]. The variable descriptions and their associated sizes are described in Table 5.1.

- $||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2$ is the regression loss and learns the metrics indirectly through the mapping

[6] Details on construction of $\boldsymbol{Q}$ given in section 5.4.3

[7] The L_1 penalty has been converted to a closed-form.

Table 5.1: Description and associated sizes of matrix variables.

Variables	Size	Description
$\boldsymbol{X}$	$n \times p$	Design Matrix
$\boldsymbol{D}$	$k \times p$	Dictionary Elements
$\boldsymbol{V}$	$n \times k$	Virtual Points
$\boldsymbol{L}$	$p \times k$	Metric s.t. $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$
$\boldsymbol{Q}$	$n \times k$	Label Consistency Matrix
$\boldsymbol{A}$	$k \times k$	Transforms $\boldsymbol{V} \rightarrow \boldsymbol{Q}$

from instances to virtual points $\boldsymbol{L} : \boldsymbol{X} \rightarrow \boldsymbol{V}$.

- $||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2 + \lambda_{\boldsymbol{V}} \sum_{\boldsymbol{v} \in \boldsymbol{V}} ||\boldsymbol{v}||_1$ is the learning of the virtual points $\boldsymbol{V}$ as a sparse representation of the instances $\boldsymbol{X}$ through the dictionary $\boldsymbol{D}$. This encourages compatibility between the virtual points and the metric.

- $||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2$ enforces *label consistency* of the virtual points and encourages that the virtual points ($\boldsymbol{V}$) are discriminative.

An interpretation of this model is that of learning a distance metric and sparse representation of the data, a combination which has recently garnered much success in other fields, most notably deep learning [6, 31] .

5.4.3 Algorithm and Optimization

The aim of our algorithm and optimization routine is to minimize equation 5.8, which is in general non-convex. We observe that it exhibits a *block multi-convex* structure with $\boldsymbol{V}$, $\boldsymbol{L}$ and $\boldsymbol{A}$ representing the separate blocks. Our optimization approach is to exploit the block multi-convex structure via a block-coordinate descent style algorithm (sometimes referred to as sequential convex optimization). Our algorithm fits under the umbrella of general block-coordinate update methods for multi-convex problems and as such is able to converge to a local minimum [82], even though some blocks ($\boldsymbol{V}$) are non-smooth.

We begin by constructing the sub-problem and feasible sets associated with each of the block variables, these are shown as equations 5.9a, 5.9b and 5.9c. Note that we have

added an L_2 regularization component to the $\boldsymbol{A}$-block and $\boldsymbol{L}$-block (equations 5.9a and 5.9b respectively), this penalty serves to increase the stability and robustness of the solution.

$$\min_{\boldsymbol{A}} f(\boldsymbol{A}) = \quad \frac{1}{2}||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2 + \frac{1}{2}\lambda_{\boldsymbol{A}}||\boldsymbol{A}||_2^2 \tag{5.9a}$$

$$\min_{\boldsymbol{L}} f(\boldsymbol{L}) = \quad \frac{1}{2}||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \frac{1}{2}\lambda_{\boldsymbol{L}}||\boldsymbol{L}||_2 \tag{5.9b}$$

$$\begin{aligned} \min_{\boldsymbol{V}} f(\boldsymbol{V}) = \quad & \frac{1}{2}||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \frac{1}{2}||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2 + \frac{1}{2}||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2 \\ s.t. \quad & \boldsymbol{V}_i \leq \beta_{\boldsymbol{V}} \quad \forall i \in [1, 2, ..., N] \end{aligned} \tag{5.9c}$$

Each of the subproblems are convex with respect to their corresponding block variable. We observe that two of the subproblems (equations 5.9a and 5.9b) are instances of the well-known ridge regression problem [71] and have closed-form solutions. The sub-problem corresponding with the V-block update is seen to be a set of independent lasso[70] problems, each centered on a column of $\boldsymbol{V}$.

Given the design and label matrices ($\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively), our approach is to begin by initializing the dictionary $\boldsymbol{D}$ and constructing the label discrimination matrix $\boldsymbol{Q}$. The optimization proceeds by updating the blocks $\boldsymbol{V}$, $\boldsymbol{A}$, and $\boldsymbol{L}$ in a deterministic round-robin fashion during each iteration until a termination condition is reached. Details of the initialization, construction, and updates of each of the blocks are described as follows.

5.4.3.1 $\boldsymbol{D}$ – Construction

The elements $\boldsymbol{d} \in \boldsymbol{D}$ are constructed as class-specific dictionary elements. The instances $\boldsymbol{x} \in \boldsymbol{X}$ are sorted according to their associated label value and the K-SVD algorithm [1] is applied to each set to extract the dictionary elements specific to each label. This procedure is done in such a manner that the distribution of class-specific dictionary elements is roughly equivalent to the distribution of label values. Each dictionary element $\boldsymbol{d} \in \boldsymbol{D}$ is assigned a fixed class label matching the class label of the instances it was initialized from.

5.4.3.2 $\boldsymbol{Q}$ – Construction

Given the dictionary $\boldsymbol{D}$ and the design matrix $\boldsymbol{X}$, the label discrimination matrix is constructed as a zero-one matrix encouraging *label-consistency*[38] between the virtual points $\boldsymbol{V}$. For example, given that $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{d}_1, \boldsymbol{d}_2\}$ are associated with the label A and $\{\boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{d}_3, \boldsymbol{d}_4, \boldsymbol{d}_5, \boldsymbol{d}_6\}$ are associated with label B,[8] The matrix $\boldsymbol{Q}$ would be defined as:

$$\boldsymbol{Q} \equiv \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5.4.3.3 $\boldsymbol{A}$ – Update

Update of $\boldsymbol{A}$ is done by minimization of equation 5.9a, which is converted to the unconstrained equivalent (equation 5.10) with parameter $\lambda_{\boldsymbol{A}}$ controlling the regularization strength on $\boldsymbol{A}$. Equation 5.10 is a multivariate ridge regression problem and has a known closed-form solution.

$$\boldsymbol{A} = \underset{\boldsymbol{A}}{argmin} ||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2 + \lambda_{\boldsymbol{A}} ||\boldsymbol{A}||_F^2 \tag{5.10}$$

5.4.3.4 $\boldsymbol{L}$ – Update

The update of $\boldsymbol{L}$ is calculated by solving a separate multivariate ridge regression problem (equation 5.11) in the same manner as the $\boldsymbol{A}$ update procedure.

[8] Although $\boldsymbol{Q}$ is best understood with the labels, we note that explicit knowledge of the label values is not required to construct $\boldsymbol{Q}$. All that is needed are the known groups of similar and dissimilar samples.

$$\boldsymbol{L} = \underset{\boldsymbol{L}}{argmin} ||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \lambda_{\boldsymbol{L}} ||\boldsymbol{L}||_2^2 \tag{5.11}$$

5.4.3.5 $\boldsymbol{V}$ – Update

Given $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{Q}$, the $\boldsymbol{V}$ block is updated by minimization of equation 5.9c. We choose to minimize this equation by application of the Alternating Direction Method of Multipliers [10] (ADMM), which takes the general form of equation 5.12.

$$\begin{aligned} \min_{\boldsymbol{V}} \quad & F(\boldsymbol{V}, \boldsymbol{Z}) \quad = f(\boldsymbol{V}) + g(\boldsymbol{Z}) \\ s.t. \quad & \boldsymbol{V} - \boldsymbol{Z} = 0 \end{aligned} \tag{5.12}$$

Taking the V-update problem shown as equation 5.9c, we let $f(\boldsymbol{V}) = \frac{1}{2}||\boldsymbol{X}\boldsymbol{L} - \boldsymbol{V}||_F^2 + \frac{1}{2}||\boldsymbol{X} - \boldsymbol{V}\boldsymbol{D}||_F^2 + \frac{1}{2}||\boldsymbol{Q} - \boldsymbol{V}\boldsymbol{A}||_F^2$ and $f(\boldsymbol{Z}) = \frac{1}{2}\lambda_{\boldsymbol{V}}||\boldsymbol{V}||_1$ to arrive at the problem-specific ADMM formulation. Minimization of equation 5.9c involves a three-step iterative process (shown as equation 5.13) updating the virtual points matrix $\boldsymbol{V}$ and the associated dual variables $\boldsymbol{Z}$ at each step k. In equation 5.13, the step size is determined by ρ and the matrix $\boldsymbol{U}$ is an intermediate variable.

$$\begin{aligned} \boldsymbol{V}^{k+1} \quad & \leftarrow \underset{\boldsymbol{V}}{\text{argmin}} \quad \left(f(\boldsymbol{V}) + \frac{\rho}{2} ||\boldsymbol{V} - \boldsymbol{Z}^k + \boldsymbol{U}^k||_F^2 \right) \\ \boldsymbol{Z}^{k+1} \quad & \leftarrow \Pi_{\frac{\lambda_V}{\rho}} \left(\boldsymbol{X}^{k+1} + \boldsymbol{U}^k \right) \\ \boldsymbol{U}^{k+1} \quad & \leftarrow \boldsymbol{U}^k + \boldsymbol{V}^{k+1} - \boldsymbol{Z}^{k+1} \end{aligned} \tag{5.13}$$

The ADMM $\boldsymbol{V}$-update consists of a separate smooth convex minimization problem. Setting the gradient of each component of $f(\boldsymbol{V})$ to zero produces a linear equation (eq. 5.14) with a closed-form solution. In our implementation we solve this equation by application of the Cholesky decomposition ($\frac{1}{3}n^3$ FLOPS) to the left hand side, followed by forward and backward substitutions (n^2 FLOPS each). The left hand side of equation 5.14 does not change between iterations. This means we can save a considerable amount of computations by calculating the Cholesky decomposition on the first iterate and caching the result. All

that is needed on subsequent iterations is to load the cached factorization and apply forward and backward substitutions.

$$\left(\boldsymbol{D}\boldsymbol{D}^T + \boldsymbol{A}\boldsymbol{A}^T + (1+\rho)\boldsymbol{I}\right)\boldsymbol{V} = \boldsymbol{X}(\boldsymbol{L} + \boldsymbol{D}^T) + \boldsymbol{Q}\boldsymbol{A}^T + \rho(\boldsymbol{Z} - \boldsymbol{U})\boldsymbol{I} \tag{5.14}$$

The ADMM $\boldsymbol{Z}$-update is calculated by the application of a projection operator onto the current iterate $\boldsymbol{Z}^k$. We are ultimately interested in the creation of sparsity via L_1-regularization, the associated projection function is the *soft-thresholding operator* (equation 5.15), applied to each element of $\boldsymbol{Z}^k$.

$$\Pi_\lambda(v) = \begin{cases} v - \lambda & \text{if } v > \lambda \\ v + \lambda & \text{if } v < -\lambda \\ 0 & \text{if } v \in [-\lambda, \lambda]. \end{cases} \tag{5.15}$$

To determine algorithm convergence, we monitor the primal and dual feasibility residuals as suggested in [10]. For our specific problem, the primal feasibility residual is given as $\boldsymbol{R}^k = \boldsymbol{V}^k - \boldsymbol{Z}^k$ and the dual feasibility residual is $\boldsymbol{S}^{k+1} = \boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k$. The optimization procedure is terminated when the residuals become small enough relative to given absolute(ϵ^{abs}) and relative(ϵ^{tol}) tolerances (equation 5.16). The $\sqrt{n}$ scales the termination criteria relative to the number of samples(n).

$$\begin{aligned} ||\boldsymbol{R}^k||_F &\leq \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\max\left\{||\boldsymbol{V}^k||_F, ||\boldsymbol{Z}^k||_F\right\} \\ ||\boldsymbol{S}^k||_F &\leq \sqrt{n}\epsilon^{abs} + \epsilon^{rel}||\rho\boldsymbol{U}^k||_F \end{aligned} \tag{5.16}$$

5.4.3.6 Algorithm Summary

The main algorithm alternatively updates each block $\boldsymbol{V}, \boldsymbol{A}$ and $\boldsymbol{L}$ in a round-robin fashion. The only complex portion if the update of the $\boldsymbol{V}$-block, which requires a separate optimization routine. We provide a pseudo-code listing for this routine as Algorithm

Algorithm 1 Pseudo-code describing the $\boldsymbol{V}$-update ADMM optimization routine.

Require: $max\_iter \geq 0$, $\rho > 0$, $tol^{rel} > 0$, $tol^{abs} > 0$
Require: $\boldsymbol{D} \in \mathbb{R}^{k \times p}$ ▷ Constructed as described in Section 5.4.3.2
Require: $\boldsymbol{Q} \in \mathbb{R}^{n \times q}$ ▷ Constructed as described in Section 5.4.3.1

```
k = 0
V^k ⇐ Q
Z^k ⇐ Q
U^k ⇐ Q
while k < max_iter do
    k ⇐ k + 1
    V^{k+1} ⇐ solve eq. 5.14
    Z^{k+1} ⇐ Π_{λ_V / rho}(V^{k+1} + U^k)
    U^{k+1} ⇐ U^k + V^{k+1} − Z^{k+1}
    r^prim ⇐ ||V^{k+1} − Z^{k+1}||_F
    r^dual ⇐ ||ρ · (Z^{k+1} − Z^k)||_F
    V^k ⇐ V^{k+1}
    Z^k ⇐ Z^{k+1}
    U^k ⇐ U^{k+1}
    if ||V^k||_F > ||Z^k||_F then
        ε^prim ⇐ √n · tol^abs + tol^rel · ||V^k||_F
    else
        ε^prim ⇐ √n · tol^abs + tol^rel · ||Z^k||_F
    end if
    ε^dual ⇐ √n · tol^abs + tol^rel · ||ρ · U^k||_F
    if r^prim < ε^prim and r^dual < ε^dual then
        break                                  ▷ Optimization has converged.
    end if
end while
```

5.4.4 Theoretical Analysis and Interpretation

In this section we provide a rudimentary theoretical analysis of the proposed method. In comparison to convex models, relatively little is known about the stability or generalization error abilities of non-convex models. We proceed by analyzing the resulting subproblems defining the update of each block variable (equations 5.9a, 5.9b, 5.9c).

The L-block update (equation 5.9b) is equivalent to the loss function in [59] and is consistent under the uniform stability framework of [9], we reproduce the theorem here for completeness. In the below theorems $R(\boldsymbol{L})$ represents the true risk and $\hat{R}(\boldsymbol{L})$ represents the empirical risk[9].

Theorem 5.4.1. *Let $||\boldsymbol{v}||_2 \leq C_{\boldsymbol{v}}$ for any $\boldsymbol{v} \in \mathcal{V}$ and $||\boldsymbol{x}||_2 \leq C_{\boldsymbol{x}}$ for any $\boldsymbol{x} \in \mathcal{X}$. With probability $1-\delta$, for matrix $\boldsymbol{L}$ optimal solution of L-block update (eq. 5.9b) we have:*

$$\begin{aligned} R(\boldsymbol{L}) \leq \hat{R}(\boldsymbol{L}) + \frac{8C_v^2C_x^2}{n\lambda_L}\left(1+\frac{C_x}{\sqrt{\lambda_A}}\right)^2 \\ \left(\frac{16C_x^2}{\lambda_A}+1\right)C_v^2\left(1+\frac{C_x}{\sqrt{\lambda_L}}\right)^2\sqrt{\frac{ln(\frac{1}{\delta})}{2n}} \end{aligned} \tag{5.17}$$

The A-block update (equation 5.9a) yields a similar consistency result(Theorem 5.4.2) via a direct application of Theorem 5.4.1.

Theorem 5.4.2. *Let $||\boldsymbol{q}||_2 \leq C_{\boldsymbol{q}}$ for any $\boldsymbol{q} \in \mathcal{Q}$ and $||\boldsymbol{v}||_2 \leq C_{\boldsymbol{v}}$ for any $\boldsymbol{v} \in \mathcal{V}$. With probability $1-\delta$, for matrix $\boldsymbol{A}$ optimal solution of A-block update (eq. 5.9b) we have:*

$$\begin{aligned} R(\boldsymbol{A}) \leq \hat{R}(\boldsymbol{A}) + \frac{8C_q^2C_v^2}{n\lambda_A}\left(1+\frac{C_v}{\sqrt{\lambda_A}}\right)^2 \\ \left(\frac{16C_v^2}{\lambda_A}+1\right)C_q^2\left(1+\frac{C_v}{\sqrt{\lambda_A}}\right)^2\sqrt{\frac{ln(\frac{1}{\delta})}{2n}} \end{aligned} \tag{5.18}$$

[9]For more details see [59]

It is well known that sparsity and algorithmic stability contradict each other [81], leading us to believe that the V-Block update is not consistent under the uniform stability framework [9]. Theoretical guarantees of this update step are left for future work.

Referring back to the connection between regressive and classic metric learning (Theorem 5.2.1), we observe that the ideal margin amounts γ_1 and γ_{-1} in the classic formulation are influenced by the virtual point placement. Our interpretation of the proposed method is that in the classic sense, each update of $\boldsymbol{V}$ may be viewed as a dynamic adjustment of the ideal margin amounts γ_1 and γ_{-1}.

Table 5.2: Summary of data characteristics. The citations associated with each dataset denote where the data may be downloaded and the original work (if known).

Dataset	**# Samples**	**# Features**	**# classes**
Balance [68, 49]	625	4	3
Credit [62, 49]	690	46	2
Digits [42, 58]	1797	64	10
German [49]	1000	61	2
Segment [49]	210	19	7
Wisconsin [56, 58]	569	30	2

5.5 Experiments

We conducted an experiment to demonstrate the effectiveness of our proposed algorithm in a classification scenario. We also provide a useful visualization illustrating the relative placement of instances and their respective virtual points.

5.5.1 Classification Experiment

In the classification experiment, we selected datasets from a variety of domains including measurement (Balance), imaging (Segmentation), finance (Credit, German), and biology/medicine (Wisconsin). The datasets used in our experiments may be found in either the UCI Machine Learning Repository [49] or as part of the scikit-learn python package [58]. A summary of the dataset characteristics including samples, features and size of the label set are arranged in Table 5.2.

In our evaluation we used the K-Nearest Neighbors [16] algorithm (using the Euclidean distance) as a simple baseline measure. The Large-Margin Nearest Neighbors (LMNN) [77] approach is arguably the most popular of the margin-based metric learning approaches and serves as a good representation of methods of this type. Finally, we also compare with the current state of the art in Regressive Virtual Metric Learning (RVML) [59]. Including two styles of RVML which use different methods of selecting/learning the virtual points. RVML-class uses a class-based representation of the virtual points and RVML-transport learns the virtual points as a problem of optimal transport.

In the classification experiment we randomly divided the data into 60% training, 20% validation, and 20% testing. The training and validation sets are used for parameter tuning and estimation of training error. Once hyper-parameters are tuned, the model is trained on the combined training and validation sets, and predictions made on the testing set. Separating the model selection and model evaluation procedures in this manner is a well-known technique to help prevent any biasing of the model evaluation by the model selection process [13].

For all algorithms, we classified new instances by projecting them into the learned space and using k-NN with $k = 3$. We refer to our proposed algorithm as Regressive Virtual Metric Learning with Dynamic Margins (RVML-DM). For RVML-class and RVML-transport, the regularization parameter λ was searched over the set $\lambda \in 1 \times 10^k$, where $k \in \{-6, -5, ..., 3\}$. For RVML-DM we set $\lambda_L = 1\text{e}{-5}$ and $\lambda_A = 1\text{e}{-5}$. We tune the regularization parameter on the virtual points $\lambda_V \in 1 \times 10^k$, where $k \in \{-6, -5, ..., -1\}$. The LMNN algorithm has no regularization hyper-parameter available to tune.

All algorithms in our experiment were implemented in Python. We used the implementation of k-NN available in the scikit-learn python package. For the LMNN algorithm, we used the popular python implementation called PyLMNN[10]. The implementation of RVML-class and RVML-transport was obtained from the authors of [59][11] and modified slightly to fit our experimental setup. Finally, we implemented our algorithm using Python, with the optimization portions in C++. In the spirit of reproducible research, we make our implementation available for download[12]. All experiments were executed on a 14-core Intel Xeon E5 series processor with 64 GB of memory.

We report the classification experiments performance results using the F1 score, which is the harmonic mean of precision and recall. The F1-score was chosen as the production of a good F1-score also requires that the algorithm has good precision and recall scores. In the

[10]Available at: `http://pylmnn.readthedocs.io/en/latest/index.html`

[11]Available at: `https://perso.univ-st-etienne.fr/pem82055/RegressiveVirtualMetricLearning.html`

[12]www.github.com/jstamand

case of multi-class data we report the micro-averaged F1-score, which is the harmonic mean of the F1-score from each class label.

5.5.2 Visualization Experiment

The location of the virtual points and their relationship to the location of the samples is unclear when viewing the data directly. We conducted some experiments to visualize the relationship between instances and their respective virtual points. We took the best hyper-parameters for each algorithm as found in the classification experiment to train each algorithm on each dataset, the learned embedding function associated with each metric was extracted and used to project the samples and virtual points into the embedding space.

We reduced the dimension of the embedding space using the T-Stochastic Neighborhood Embedding (T-SNE) [72] algorithm. T-SNE is a probabilistic technique which aims to preserve the similarity between each pair of points and is extremely popular for the generation of visualizations. The results for the experiment are shown in section 5.5.3.2.

5.5.3 Results and Discussion

5.5.3.1 Classification Results

The results of the classification experiment are shown in Table 5.3. We found that on four of the six datasets, the proposed method (RVML-DM) yielded the highest average micro F1-score. We calculated the significance of the results using Welch's two-sided t-test [78]. On the Urban dataset the results was significant on at least the 5% level and for the German dataset it was significant at least on the 1% level. Additionally, the proposed method yielded the highest F1-score when averaged across all the datasets (shown in the far right column of Table 5.3). The proposed method under performed another method on two datasets, we found that in these cases the performance of the leading method was not statistically significant, with p-values were calculated to be 0.195 (Segmentation) and 0.204 (Wisconsin).

Table 5.3: Experimental evaluation of RVML-DM vs. competing method in a classification scenario on datasets from different domains. Reported results are the micro-averaged F1-score ($F1_{Micro}$) over 20 trials. A $*$ denotes significance on the 5% level and a $**$ denotes significance on the 1% level.

Dataset	KNN	LMNN	RVML-Class	RVML-Transport	RVML-DM
Balance	0.9218	0.9446	0.9034	0.9184	**0.9484**
Credit	.6769	0.7350	0.7979	0.8092	**0.8140**
German	0.6303	0.6771	0.6471	0.6402	**0.7100****
Segmentation	0.7177	0.8536	**0.8744**	0.7798	0.8572
Urban	0.3354	0.6444	0.7120	0.6478	**0.7289***
Wisconsin	0.9055	**0.9496**	0.9057	0.9086	0.9450
Ave.F1$_{\text{Micro}}$	.6978	.8007	.8067	.7840	**.8340**

We also observe that when the proposed method was not the best performing, it was always the second best.

5.5.3.2 Visualization Results

The results of the visualization experiment are shows as Figures 5.1 (Balance), 5.2 (Credit), 5.3 (German), 5.4 (Segment), 5.5 (Urban), 5.6 (Wisconsin).

We observe in the experiments on the Urban and Wisconsin datasets (Figures 5.5 and 5.6 respectively), that RVML-DM appears to be better at separating the clusters of instances. This is particularly apparent when observing the results on the Urban datasets, where we can clearly see the classes are much better separated than the competing methods. We also note that there is also a much clearer separation of the clusters in the Wisconsin dataset. Although the performance of RVML-DM was not the best on the Segmentation dataset, there is still a clear effort in Figure 5.4 to separate the instances.

We made some interesting observations on the Credit and German datasets, (Figures 5.2 and 5.3 respectively). On the German dataset we see that the samples from the different classes are well-mixed and form some type of long and thin ribbon structure. This means that the samples may smoothly vary between several different modes, making it a particularly challenging task. The RVML-class and RVML-transport methods produce virtual points

which are all located in a tight cluster, possibly containing a single mode. In our proposed method RVML-DM, we see that the learned virtual points themselves form a similar ribbon-like structure. These virtual points are clearly much more effective at separating the two classes. We hypothesis that the virtual points could have learned to smoothly vary between different modes in a way similar to how the data is represented.

Finally, we observe that in the results on the credit dataset (Figure 5.2), for the RVML-class and RVML-transport methods, there is a small cluster of instances isolated from the rest of the instances. One explanation for this is that the instances demonstrate two distinct modes. It can be seen that the RVML-class dataset has trouble representing this as the virtual points are fixed. The RVML-transport method and RVML-DM both produce more interesting virtual points. RVML-transport produces a small separate section of red virtual points on the far left, and the separate cluster of instances is still present. For RVML-DM, there is learned cluster of blue virtual points (on the far right) which is clearly separated from the rest. Interestingly, the separate cluster of instances appears to have begun to merge with the rest. One explanation for this is that the RVML-DM method was better able to pick up on the two-mode nature of the dataset and that is reflected in the learned virtual points.

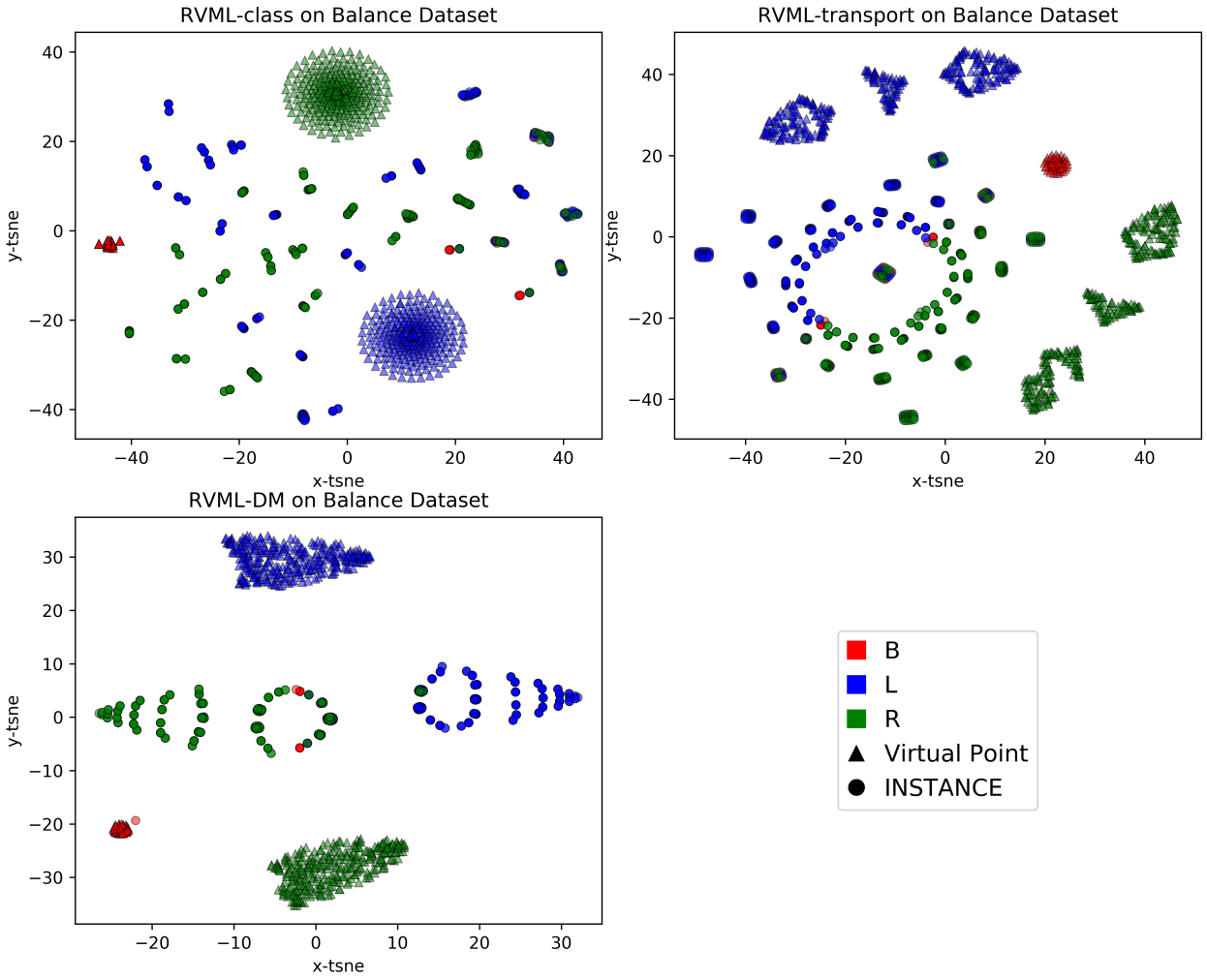

Figure 5.1: Visualization of virtual points and instances in the space induced by the learned metric. Reduction to two dimensions using T-SNE. Tunable parameters for each method chosen as those yielding the best performance in the classification experiment. Colors indicate class label and shapes denote instance / virtual point. All points drawn with partial transparency for better visualization of points in overlapping clusters. Figure best viewed in color.

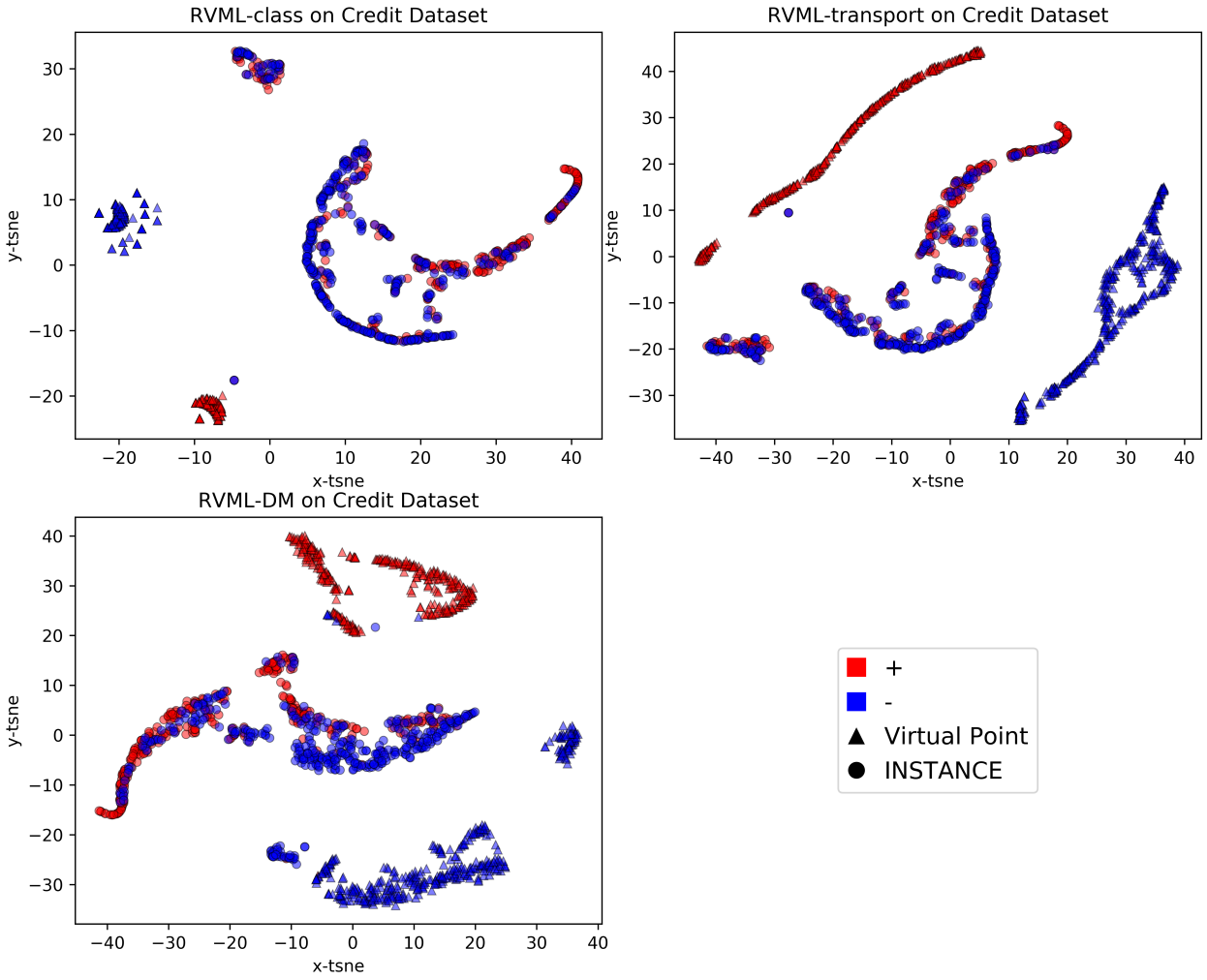

Figure 5.2: Visualization of virtual points and instances in the space induced by the learned metric. Reduction to two dimensions using T-SNE. Tunable parameters for each method chosen as those yielding the best performance in the classification experiment. Colors indicate class label and shapes denote instance / virtual point. All points drawn with partial transparency for better visualization of points in overlapping clusters. Figure best viewed in color.

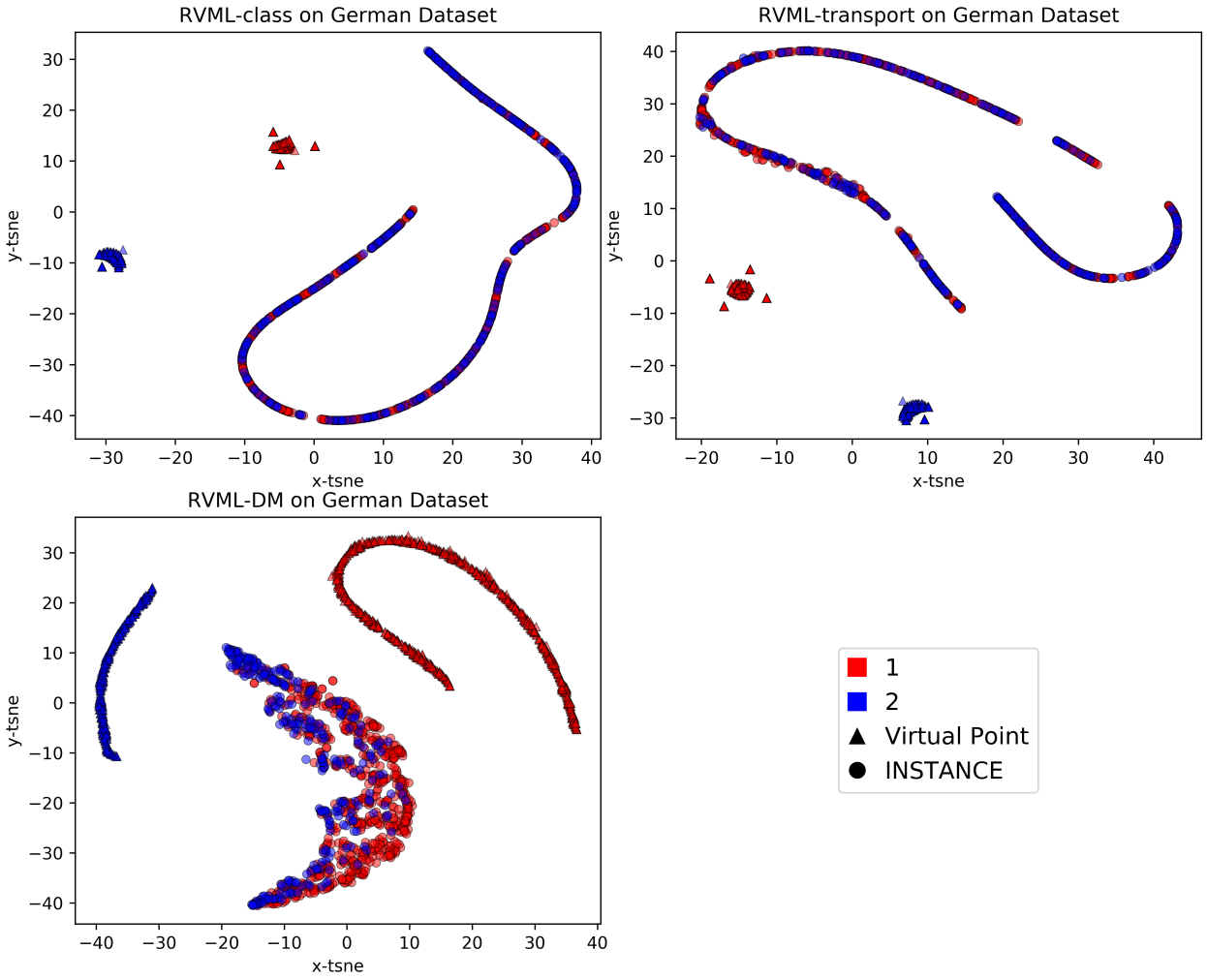

Figure 5.3: Visualization of virtual points and instances in the space induced by the learned metric. Reduction to two dimensions using T-SNE. Tunable parameters for each method chosen as those yielding the best performance in the classification experiment. Colors indicate class label and shapes denote instance / virtual point. All points drawn with partial transparency for better visualization of points in overlapping clusters. Figure best viewed in color.

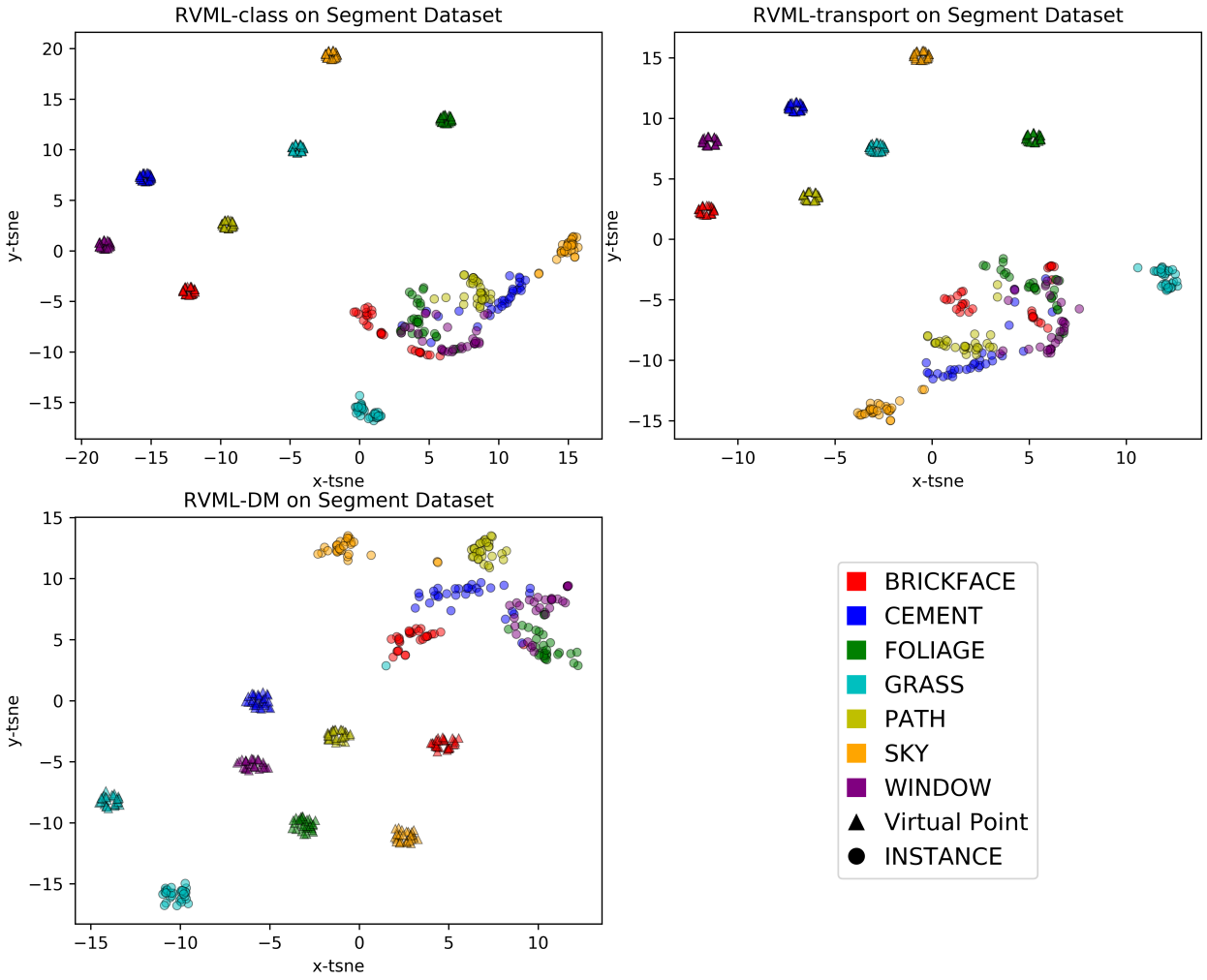

Figure 5.4: Visualization of virtual points and instances in the space induced by the learned metric. Reduction to two dimensions using T-SNE. Tunable parameters for each method chosen as those yielding the best performance in the classification experiment. Colors indicate class label and shapes denote instance / virtual point. All points drawn with partial transparency for better visualization of points in overlapping clusters. Figure best viewed in color.

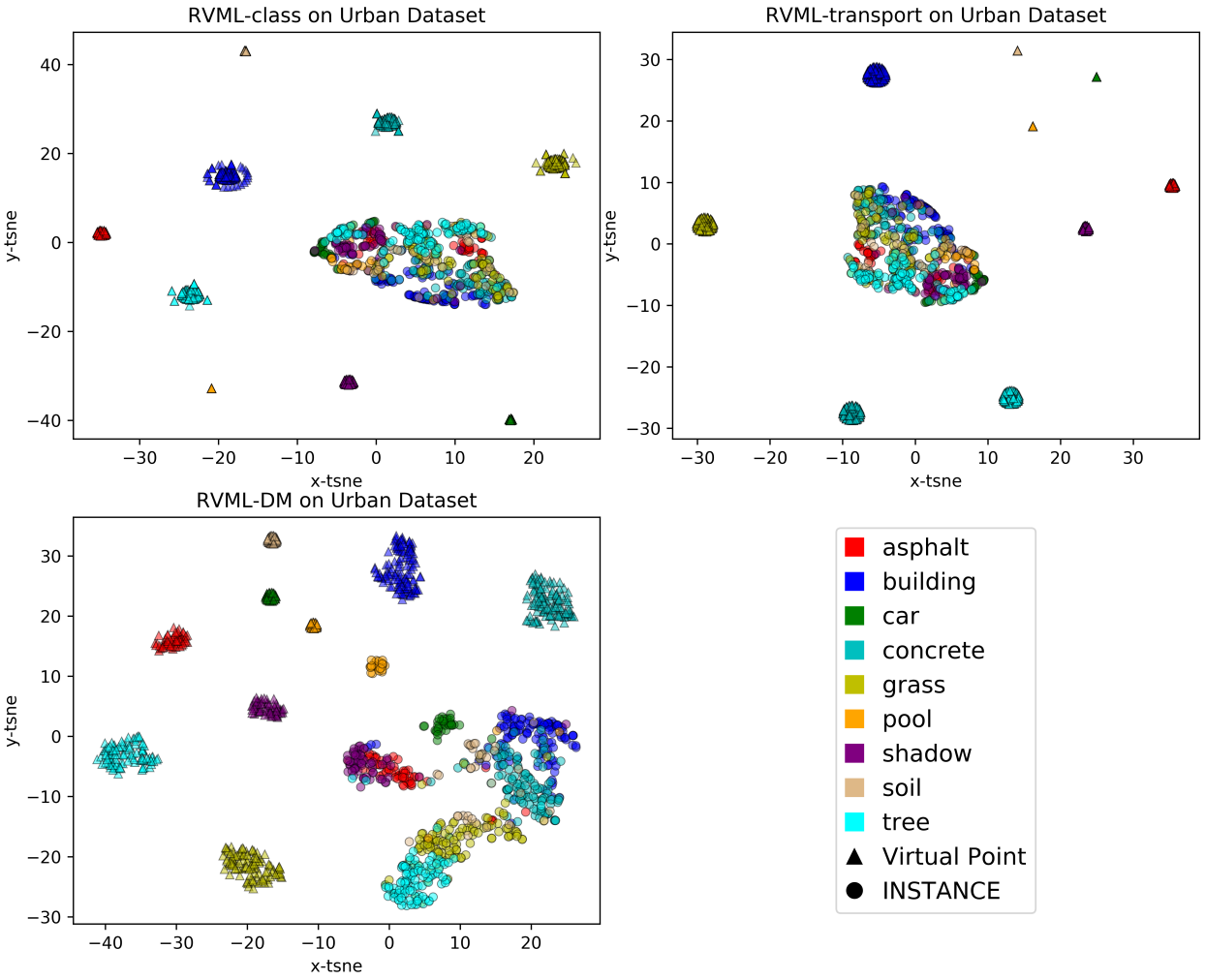

Figure 5.5: Visualization of virtual points and instances in the space induced by the learned metric. Reduction to two dimensions using T-SNE. Tunable parameters for each method chosen as those yielding the best performance in the classification experiment. Colors indicate class label and shapes denote instance / virtual point. All points drawn with partial transparency for better visualization of points in overlapping clusters. Figure best viewed in color.

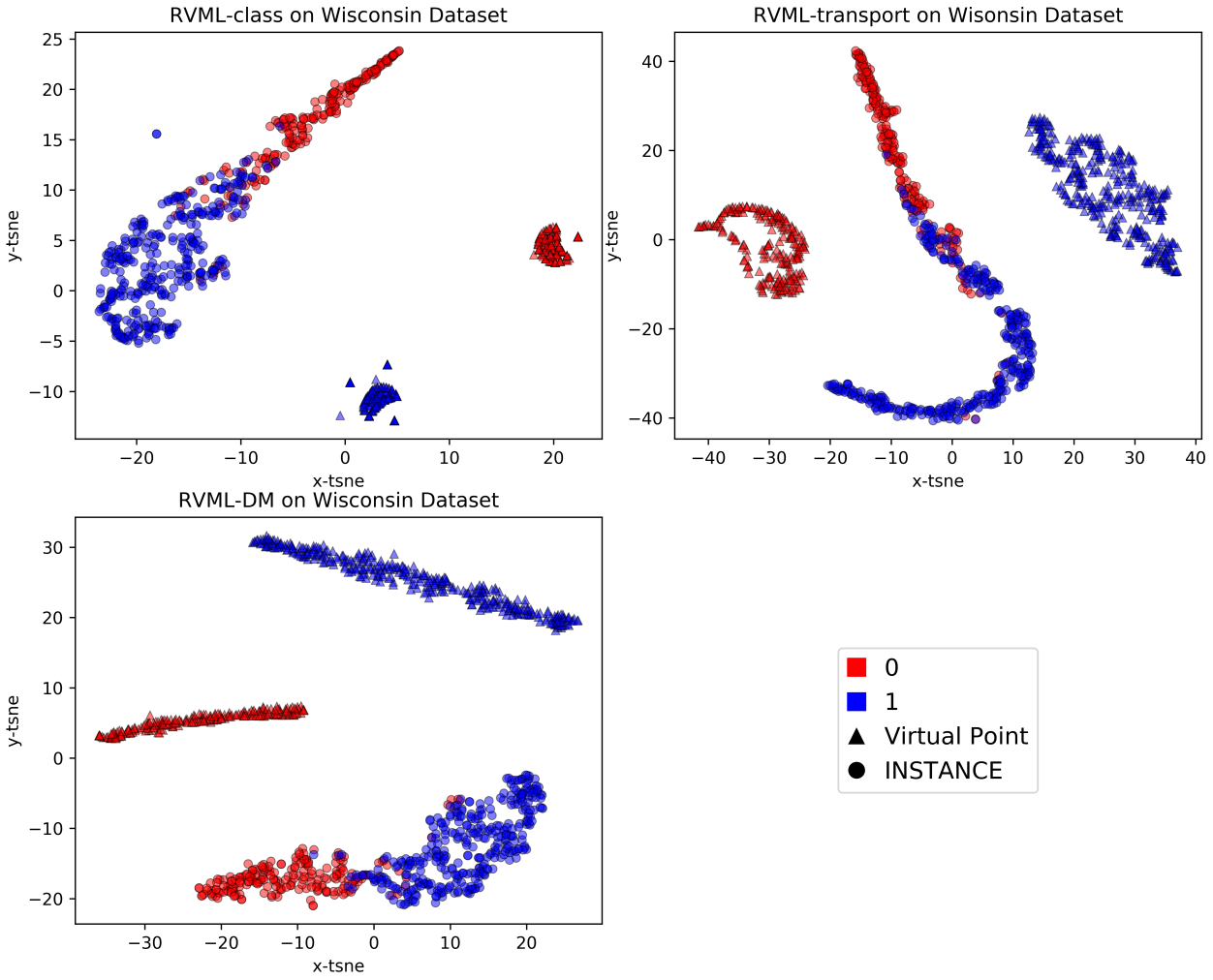

Figure 5.6: Visualization of virtual points and instances in the space induced by the learned metric. Reduction to two dimensions using T-SNE. Tunable parameters for each method chosen as those yielding the best performance in the classification experiment. Colors indicate class label and shapes denote instance / virtual point. All points drawn with partial transparency for better visualization of points in overlapping clusters. Figure best viewed in color.

5.6 Conclusion and Future Work

We proposed a new method for discriminative regressive virtual metric learning where the goal is to promote discriminability and compatibility of the virtual points with the metric. We demonstrated through several experiments that our method is capable of outperforming the current state of the art on 4/6 datasets from a variety of domains. In the visualization experiment we used T-SNE to cluster the embedded data in a two dimensional space. We found that RVML-DM yielded transformed instances and virtual points which were easy to cluster, while RVML-class and RVML-transport produced results which were unable to be fully disentangled. In comparison to the well-established baseline LMNN, the proposed method is much faster and yields better performance on average. While in comparison with the state of the art in regressive metric learning, our method falls in the middle of the road in terms of scalability, while providing a moderate increase in performance.

Our plans for future work in this direction is to further enhance the scalability of RVML-DM through more efficient optimization and establish additional theory evaluating the stability and robustness of our algorithm from a global perspective. Additionally, we believe that a method for learning the virtual points under the assumption of multi-modal data could be a good future research direction.

5.7 Appendix

In the interest of comparing the proposed RVML-DM algorithm with the work of the previous chapter, we ran an additional experiment under the same conditions and data as that in section 5.5.1. The results of this experiment are shown in Table 5.4.

Table 5.4: Summary of experimental results, displayed metric is the micro-F1 score averaged over 20 train/validation/test splits of the data. The best results as measured by the F1-score (which is micro-averaged in the case of multi-class data) for each dataset are highlighted in **bold**. A $**$ denotes a p-value on the 1% significance level ($p < 0.01$).

Method	Balance	Credit	German	Segmentation	Urban	Wisconsin
KNN	0.9218	0.6769	0.6303	0.7177	0.3354	0.9055
LMNN	0.9446	0.7350	0.6771	0.8536	0.6444	**0.9496**
RVML-Class	0.9034	0.7979	0.6471	**0.8744**	0.7120	0.9057
RVML-Transport	0.9184	0.8092	0.6402	0.7798	0.6478	0.9086
RRVMLSS-Class	0.9052	0.7922	0.6215	0.8738	**0.7372**	0.9081
RRVMLSS-Transport	0.9258	0.8096	0.6143	0.8077	0.6602	0.9184
RVML-DM	**0.9484**	**0.8140**	**0.7100****	0.8572	0.7289	0.9450

Chapter 6

Conclusion and Future Work

We presented three algorithms for distance metric learning under two different general strategies: the large-margin approach, and the new regressive virtual metric learning approach. Under the large-margin strategy, we developed a new method for learning a distance metric in a high-dimensional feature space. We specifically addressed the case of high-dimensional heterogeneous data, where different portions of the input space could have different dominating factors. Our developed metric took a local approach and was composed of both global and local components. We successfully applied this technique to a distance metric learning problem with over 1 billion parameters to learn.

Under the regressive virtual metric learning approach, we proposed and developed two improvements. The first improvement was the addition of structured sparsity to the distance metric which allowed it to reject noisy, corrupted, or irrelevant inputs by relying on a few select components. We showed under a classification scenario that our structured sparse approach to regressive distance metric learning was more effective at learning in the presence of irrelevant features. Although we learned the metric in an indirect manner, our structured sparse approach resulted in a sparse metric, which was more interpretable than the competing approaches.

We investigated the influence of virtual points in the regressive virtual metric learning scenario. We discussed two ideas we believed to be important when considering virtual points: (1) *virtual point compatability*, and (2) *virtual point discrimination potential*. Based on these ideas we proposed a new objective which is able to learn the virtual points and the metric jointly. We showed a connection to sparse dictionary learning and found that by selecting the dimension of the sparse representation, we can control the output sparsity(rank) of the distance metric. Finally, we put forth an interpretation of our approach, that it is similar to learning a distance metric under the large-margin approach, where the margins are dynamically adjusted. This method proved very effective and we showed that it was able to outperform competing methods on datasets from several different domains.

In our first distance metric learning approach, we learned a high-dimensional local metric. Due to the structured sparse updates in this approach, the memory access patterns are particularly scattered. We wonder if there is a more efficient approach to running our algorithm with more effective memory access patterns. This would enable our method to run on a Graphics Processing Unit (GPU). GPUs have become the de-facto hardware for deep learning and are now commonly found in many devices. The ability to run our algorithm on this type of device would allow us to scale to datasets with even higher dimensions.

In conclusion, we believe that a combination of our last two approaches on the regressive virtual metric learning problem could yield further improvements in performance. In addition, we are specifically interested in different ways of learning the virtual points. In this dissertation we explicitly considered learning the virtual points as structured sparse linear representation of the data. An interesting direction for future research would be to consider ways of integrating nonlinear features into the representation. In particular, we would like to consider whether or not a deep neural network could be used to extract nonlinear features which can be leveraged by a distance metric.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

[2] C. Bao, H. Ji, Y. Quan, and Z. Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1356–1369, July 2016.

[3] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, march-april 2011.

[4] A. Bellet, A. Habrard, and M. Sebban. Similarity learning for provably accurate sparse linear classification. *ArXiv e-prints*, June 2012.

[5] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, aug 2013.

[7] Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pages 217–235, London, UK, UK, 1999. Springer-Verlag.

[8] R. Binetti, F. M. Costamagna, and I. Marcello. Exponential growth of new chemicals

and evolution of information relevant to risk control. *Ann. Ist. Super. Sanita*, 44(1):13–15, 2008.

[9] Olivier Bousquet and André Elisseeff. Stability and generalization. *J. Mach. Learn. Res.*, 2:499–526, Mar 2002.

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan 2011.

[11] Kenneth P. Burnham and David R. Anderson. Multimodel Inference: Understanding AIC and BIC in Model Selection. *Sociological Methods & Research*, 33(2):261–304, 2004.

[12] Deng Cai, Xuanhui Wang, and Xiaofei He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 105–112, New York, NY, USA, 2009. ACM.

[13] Gavin C. Cawley and Nicola L.C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, 11:2079–2107, August 2010.

[14] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[16] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, September 2006.

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In

Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, pages 2292–2300, USA, 2013. Curran Associates Inc.

[18] Jason V. Davis and Inderjit S. Dhillon. Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 195–203, New York, NY, USA, 2008. ACM.

[19] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216, New York, NY, USA, 2007. ACM.

[20] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73:415–34, 2012 Feb 9 2012.

[21] Ethan Fetaya and Shimon Ullman. Learning local invariant mahalanobis distances. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 162–168. JMLR Workshop and Conference Proceedings, 2015.

[22] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

[23] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pages 1–8, 2007.

[24] Xingyu Gao, Steven C. H. Hoi, Yongdong Zhang, Ji Wan, and Jintao Li. SOML: sparse online metric learning with application to image retrieval. In *Proceedings of the*

Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada., pages 1206–1212, 2014.

[25] Amir Globerson and Sam T. Roweis. Metric learning by collapsing classes. In *NIPS*, pages 451–458, 2005.

[26] Jacob Goldberger, Geoffrey E Hinton, Sam T. Roweis, and Ruslan R Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2005.

[27] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[28] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press, 2006.

[29] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the nips 2003 feature selection challenge. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT Press, 2005.

[30] Clara Higuera, Katheleen J. Gardiner, and Krzysztof J. Cios. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLOS ONE*, 10(6):1–28, 06 2015.

[31] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog, editors, *SIMBAD*, volume 9370 of *Lecture Notes in Computer Science*, pages 84–92. Springer, 2015.

[32] Yi Hong, Quannan Li, Jiayan Jiang, and Zhuowen Tu. Learning a mixture of sparse distance metrics for classification and dimensionality reduction. In *2011 International Conference on Computer Vision*, pages 906–913, Nov 2011.

[33] Kaizhu Huang, Rong Jin, Zenglin Xu, and Cheng-Lin Liu. Robust metric learning by smooth optimization. *CoRR*, abs/1203.3461, 2012.

[34] Zhouyuan Huo, Feiping Nie, and Heng Huang. Robust and effective metric learning using capped trace norm: Metric learning via capped trace norm. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1605–1614, New York, NY, USA, 2016. ACM.

[35] Sung Ju Hwang, Kristen Grauman, and Fei Sha. *Learning a tree of metrics with disjoint visual features*. 2011.

[36] Martin Jaggi. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zurich, Oct 2011.

[37] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 427–435. JMLR Workshop and Conference Proceedings, 2013.

[38] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664, Nov 2013.

[39] D. M. Johnson, C. Xiong, and J. J. Corso. Semi-Supervised Nonlinear Distance Metric Learning via Forests of Max-Margin Cluster Hierarchies. *ArXiv e-prints*, February 2014.

[40] Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization

with frank-wolfe algorithm. In *European Conference on Computer Vision*, pages 253–268. Springer, 2014.

[41] Purushottam Kar and Prateek Jain. Similarity-based learning via data driven embeddings. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 1998–2006, USA, 2011. Curran Associates Inc.

[42] C Kaynak. Methods of combining multiple classifiers and their applications to handwritten digit recognition. Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University, 1995.

[43] Dor Kedem, Stephen Tyree, Fei Sha, Gert R. Lanckriet, and Kilian Q Weinberger. Nonlinear metric learning. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2573–2581. Curran Associates, Inc., 2012.

[44] Yumi Kondo, Matias Salibian-Barrera, and Ruben Zamar. Rskc: An r package for a robust and sparse k-means clustering algorithm. *Journal of Statistical Software, Articles*, 72(5):1–26, 2016.

[45] Brian Kulis. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

[46] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 496–504, Cambridge, MA, USA, 2015. MIT Press.

[47] Simon Lacoste-Julien, Fredrik Lindsten, and Francis Bach. Sequential kernel herding: Frank-wolfe optimization for particle filtering. *arXiv preprint arXiv:1501.02056*, 2015.

[48] Hao Lei, Kuizhi Mei, Jingmin Xin, Peixiang Dong, and Jianping Fan. Hierarchical learning of large-margin metrics for large-scale image classification. *Neurocomputing*, 208:46 – 58, 2016. SI: BridgingSemantic.

[49] M. Lichman. UCI machine learning repository, 2013.

[50] Daryl Lim, Gert R. G. Lanckriet, and Brian McFee. Robust structural metric learning. In *ICML (1)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 615–623. JMLR.org, 2013.

[51] Daryl K. H. Lim and Gert Lanckriet. Efficient learning of mahalanobis metrics for ranking. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1980–II–1988. JMLR.org, 2014.

[52] Kuan Liu, Aurélien Bellet, and Fei Sha. Similarity Learning for High-Dimensional Sparse Data. In *AISTATS*, 2015.

[53] Wei Liu, Cun Mu, Rongrong Ji, Shiqian Ma, John R. Smith, and Shih-Fu Chang. Low-rank similarity metric learning in high dimensions. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2792–2799. AAAI Press, 2015.

[54] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006.

[55] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, April 1936.

[56] Olvi L. Mangasarian, W. Nick Street, and William H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

[57] Brian Mcfee and Gert Lanckriet. Metric learning to rank. In *In Proceedings of the 27th annual International Conference on Machine Learning (ICML*, 2010.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[59] Michaël Perrot and Amaury Habrard. Regressive virtual metric learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1801–1809. Curran Associates, Inc., 2015.

[60] A. M. Qamar and E. Gaussier. Online and batch learning of generalized cosine similarities. In *2009 Ninth IEEE International Conference on Data Mining*, pages 926–931, Dec 2009.

[61] A. M. Qamar, E. Gaussier, J. P. Chevallet, and J. H. Lim. Similarity learning for nearest neighbor classification. In *2008 Eighth IEEE International Conference on Data Mining*, pages 983–988, Dec 2008.

[62] J. R. Quinlan. Simplifying decision trees. *Int. J. Man-Mach. Stud.*, 27(3):221–234, September 1987.

[63] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

[64] Shreyas Saxena and Jakob Verbeek. Coordinated Local Metric Learning. In *ICCV ChaLearn Looking at People workshop*, Proceedings IEEE International Conference on Computer Vision Workshops, pages 369–377, Santiago, Chile, December 2015. IEEE.

[65] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In S. Thrun, L. K. Saul, and P. B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 41–48. MIT Press, 2004.

[66] Chunhua Shen, Junae Kim, Lei Wang, and Anton Van Den Hengel. Positive semidefinite metric learning using boosting-like algorithms. *J. Mach. Learn. Res.*, 13(1):1007–1036, April 2012.

[67] Yuan Shi, Aurélien Bellet, and Fei Sha. Sparse compositional metric learning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 2078–2084. AAAI Press, 2014.

[68] Thomas R. Shultz, Denis Mareschal, and William C. Schmidt. Modeling cognitive development on balance scale phenomena. *Machine Learning*, 16(1):57–86, Jul 1994.

[69] Joseph St.Amand and Jun Huan. Sparse compositional local metric learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1097–1104, New York, NY, USA, 2017. ACM.

[70] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[71] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

[72] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[73] Cédric Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer, 2009 edition, Sep 2008.

[74] Jun Wang, Adam Woznica, and Alexandros Kalousis. Parametric local metric learning for nearest neighbor classification. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 1601–1609, USA, 2012. Curran Associates Inc.

[75] Kilian Q Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480. MIT Press, 2006.

[76] Kilian Q. Weinberger and Lawrence K. Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1160–1167, New York, NY, USA, 2008. ACM.

[77] K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

[78] B L Welch. The Generalization of 'Student's' Problem when Several Different Population Variances are Involved. *Biometrika*, 34(1/2):28–35, 1947.

[79] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 521–528. MIT Press, 2003.

[80] Caiming Xiong, David Johnson, Ran Xu, and Jason J. Corso. Random forests for metric learning with implicit pairwise position dependence. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 958–966, New York, NY, USA, 2012. ACM.

[81] Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1801–1808. Curran Associates, Inc., 2009.

[82] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

[83] Wenzhuo Yang and Huan Xu. A unified robust regression model for lasso-like algorithms. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 585–593, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[84] B. Yao, Z. Zhao, and K. Liu. Metric learning with trace-norm regularization for person re-identification. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2442–2446, Oct 2014.

[85] D. Yao, P. Zhao, C. Yu, H. Jin, and B. Li. Sparse online relative similarity learning. In *2015 IEEE International Conference on Data Mining*, pages 529–538, Nov 2015.

[86] Han-Jia Ye, De-Chuan Zhan, Xue-Min Si, and Yuan Jiang. Learning feature aware metric. In Robert J. Durrant and Kee-Eung Kim, editors, *Proceedings of The 8th Asian Conference on Machine Learning*, volume 63 of *Proceedings of Machine Learning Research*, pages 286–301, The University of Waikato, Hamilton, New Zealand, 16–18 Nov 2016. PMLR.

[87] Yiming Ying, Kaizhu Huang, and Colin Campbell. Sparse metric learning via smooth optimization. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2214–2222. Curran Associates, Inc., 2009.

[88] Yiming Ying and Peng Li. Distance metric learning with eigenvalue optimization. *J. Mach. Learn. Res.*, 13:1–26, January 2012.

[89] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 68:49–67, 2006.

[90] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX*

Conference on Hot Topics in Cloud Computing, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

[91] De-Chuan Zhan, Ming Li, Yu-Feng Li, and Zhi-Hua Zhou. Learning instance specific distances using metric propagation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1225–1232, New York, NY, USA, 2009. ACM.

[92] Yu Zheng, Jianping Fan, Ji Zhang, and Xinbo Gao. Hierarchical learning of multi-task sparse metrics for large-scale image classification. *Pattern Recognition*, 67:97 – 109, 2017.

[93] G. Zhong, K. Huang, and C. L. Liu. Low rank metric learning with manifold regularization. In *2011 IEEE 11th International Conference on Data Mining*, pages 1266–1271, Dec 2011.